Lucas Gonçalves Martins

# EMISSÃO DISTRIBUÍDA E EM LARGA ESCALA DE CERTIFICADOS DIGITAIS

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Computação para a obtenção do Grau de Mestre em Ciência da Computação.
Orientador: Prof. Ricardo Felipe Custódio, Dr.

Florianópolis

2013

Lucas Gonçalves Martins

**EMISSÃO DISTRIBUÍDA E EM LARGA ESCALA DE CERTIFICADOS DIGITAIS**

Esta Dissertação foi julgada aprovada para a obtenção do Título de "Mestre em Ciência da Computação", e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Florianópolis, 20 de Setembro 2013.

---

Prof. Ronaldo dos Santos Mello, Dr.
Coordenador

**Banca Examinadora:**

---

Prof. Ricardo Felipe Custódio, Dr.
Orientador

---

Prof. Eugene Francis Vinod Rebello, Dr.
Universidade Federal Fluminense

---

Prof. Ricardo Pereira e Silva, Dr.
Universidade Federal de Santa Catarina

---

Prof. Ricardo Alexandre Reinaldo de Moraes, Dr.
Universidade Federal de Santa Catarina

Dedico este trabalho aos meus pais que sempre me apoiaram nesta jornada.

## AGRADECIMENTOS

Agradeço a toda equipe do Laboratório de Segurança em Computação da Universidade Federal de Santa Catarina, sem os quais a realização deste trabalho não seria possível. Agradeço, principalmente, a equipe responsável pelo desenvolvimento do sistema proposto neste trabalho, que além de colegas de trabalho se tornaram grandes amigos. Também gostaria de fazer um agradecimento especial aos amigos Rick, Felipe, Lucas e Hendri, que passaram pelas mesmas dificuldades e sempre me incentivaram nos momentos mais difíceis. Por fim, agradeço a minha família e a minha namorada, por proporcionarem os momentos mais felizes durante o tempo de realização deste trabalho.

A humanidade é como um sistema concorrente que se aproxima de um *deadlock*.

## RESUMO

A emissão de certificados digitais em larga escala de forma confiável tem sido um dos maiores desafios para a implantação de infraestruturas de chaves públicas. As soluções existentes, frutos de modelos propostos na década de 1990, não se mostram adequadas quando utilizadas para a emissão distribuída de grande quantidade de certificados digitais através de uma autoridade certificadora. Entre os problemas destacam-se a dificuldade de se implantar de forma rápida novas instalações técnicas para atender uma demanda pontual de certificados digitais e o gargalo quanto à disponibilidade, eficiência e performance, relativos à verificação de dados dos requerentes em grandes regiões geográficas. Este trabalho propõe um novo modelo e uma arquitetura de um sistema para a emissão distribuída de certificados digitais em larga escala. Foi implementado um protótipo do sistema e realizada uma comparação desta nova arquitetura com outras existentes.



**Palavras-chave:** Infraestrutura, chave, pública, modelo, certificado, digital, sistema, gerenciador, autoridade, certificadora, registro, registradora.

## LISTA DE FIGURAS

## LISTA DE TABELAS

## LISTA DE ABREVIATURAS E SIGLAS

**SUMÁRIO**

# 1 INTRODUÇÃO

Um dos grandes desafios da autenticação no meio digital (e da segurança da informação em geral) é a criação de mecanismos que possuam níveis de segurança e usabilidade satisfatórios para o sistema de informação e comunicação com seus usuários (STRAUB, 2006). A Infraestrutura de Chaves Públicas (ICP) é uma das tecnologias mais promissoras para a criação desses mecanismos. A função de uma ICP é criar evidências que comprovem a associação entre uma entidade (e.g., pessoa, máquina, empresa, etc.) e uma chave criptográfica. Essas evidências são materializadas na forma de certificados digitais. A credibilidade sobre um certificado digital está associada às entidades e à organização da ICP que o produziu. Sendo o usuário final, aquele que utiliza o certificado, o único responsável pela decisão de confiar ou não na ICP.

Alguns países instituem ICPs governamentais, cujas entidades responsáveis são controladas por instituições de governo. Esse tipo de ICP facilita a legalização da certificação digital, pois o governo tem controle sobre quais entidades fazem parte da ICP e sobre as políticas que essas entidades devem seguir. Por outro lado, ICPs governamentais costumam ser maiores e mais complexas do que ICPs privadas, pois procuram atender às necessidades de pessoas e empresas de um país inteiro. Além disso, ICPs governamentais estão diretamente envolvidas com a soberania do seu país, pois é dada fé pública aos certificados e as serviços oferecidos por ela.

O requisito de escalabilidade é fundamental para viabilizar a implantação de grandes ICPs, como é o caso de ICPs governamentais. Esse requisito tem imposto à ICP uma série de processos complexos, difíceis de manter, e que tem levado ao fracasso inúmeras implementações de ICP. São várias as críticas. Uma delas é a inexistência de um modelo de implementação de ICP que permita a gestão distribuída em larga escala e que, ao mesmo tempo, atenda aos rigorosos requisitos de segurança inerentes a um ambiente de ICP. A definição e implementação de um Sistema Gerenciadores de Certificados Digitais (SGC) distribuído que permita a emissão de certificados digitais em larga escala não é trivial e tem exigido dos especialistas uma série de novos protocolos. Alguns desses protocolos são temas recentes de pesquisa, como são as pesquisas sobre cerimônias em ICP feitas na Royal Rolloway University (BELLA; COLES-KEMP, 2012; CARLOS et al., 2013).

**Definição 1 (Escalabilidade)** É a habilidade de um sistema computacional ou processo de se comportar de maneira esperada diante do aumento inesperado de atividade. No caso de sistemas de gerenciamento de certificados

digitais, é a habilidade de emitir qualquer demanda de certificados digitais, tal como a de um país como o Brasil. Um SGC é dito escalável se permite emitir desde algumas dezenas de certificados até milhares de certificados em alguns minutos de operação, respeitando todos os rigorosos requisitos de segurança inerentes aos sistemas desse tipo, em uma grande região geográfica.

## 1.1 DEFINIÇÃO DO PROBLEMA

O problema é como emitir certificados digitais para pessoas em grande quantidade de forma segura e confiável. Existem alguns gargalos nesse processo. Entre as atividades que exigem tempo há a geração do par de chaves criptográficas usando um dispositivo criptográfico como suporte, assinatura do certificado digital pela entidade emissora, a verificação da identidade e dos atributos do requerente do certificado e a concordância do requerente em atestar o recebimento do certificado e aceitar a responsabilidade de controle da sua chave privada. Várias dessas atividades são necessariamente realizadas por seres humanos, exigindo a definição e execução de cerimônias formais no processo de emissão dos certificados. Estima-se que são necessários de 10 a 15 minutos para a emissão de um único certificado digital. Nestes termos, e considerando um dia de trabalho de 10 horas, não mais de 50 certificados são esperados ser emitidos por uma autoridade certificadora em um dia. E o que se deseja é poder emitir milhares de certificados em um dia. Certamente, há que distribuir geograficamente o processo de emissão de certificados digitais para poder atender esta demanda. E essa distribuição, devido aos requisitos de controle das chaves criptográficas da entidade emissora, não é uma tarefa trivial.

Atualmente, existem diversas recomendações que especificam padrões para as Infraestruturas de Chaves Públicas e suas aplicações, inclusive para Sistemas Gerenciadores de Certificados Digitais. Essas recomendações focam na especificação de estruturas de dados, procedimentos e protocolos (IETF, 2013). Nenhuma delas se atem ao desafio de emissão distribuída em larga escada de certificados digitais, como é o requisito quando o objetivo é a implantação de uma ICP governamental.

Os SGCs precisam cumprir com rígidos requisitos de segurança e, ao mesmo tempo, estarem preparados para ambientes de alta demanda. Porém, a ausência de modelos detalhados tem dificultado a especificação desses sistemas, o que tem levado ao desenvolvimento de sistemas que, quando colocados em operação, não conseguem atender a demanda real de certificados.

Este trabalho, que é resultado de 10 anos de pesquisas e desenvolvi-

mento de sistemas de gerenciamento de certificados digitais no Laboratório de Segurança em Computação (LabSEC) da Universidade Federal de Santa Catarina (UFSC), apresenta um modelo geral de requisitos para a implementação de SGCs que possibilita a emissão segura, distribuída e em larga escala de certificados digitais.

## 1.2 OBJETIVOS

O objetivo deste trabalho é propor um modelo para Sistemas Gerenciadores de Certificados Digitais utilizados para a emissão distribuída em larga escala de certificados digitais para pessoas.

### 1.2.1 Objetivos Específicos

Os objetivos específicos deste trabalho incluem:

- apresentar os componentes de uma ICP e suas funcionalidades;
- definir o uso correto de Autoridades de Registro (ARs) através da especificação suas funções;
- introduzir o Agente de Registro (AGR) como parte integrante do protocolo de gerenciamento do ciclo de vida de certificados digitais;
- padronizar o acesso aos recursos utilizados por ACs e ARs, como base de dados e provedores criptográficos;
- propor mecanismos de controle de acesso compatíveis com os níveis de segurança de uma ICP;
- propor um sistema de backup seguro, baseado nas regras propostas para o controle de acesso;
- propor uma cerimônia de emissão de Certificados Digitais mais amigável para o Usuário Final;
- Avaliar trabalhos e aplicações relacionados aos sistemas utilizados por Autoridades Certificadoras e de Registro;
- Propor um modelo detalhado para Sistema Gerenciador de Certificados Digitais.

## 1.3 MOTIVAÇÃO

O Laboratório de Segurança em Computação (LabSEC), da Universidade Federal de Santa Catarina (UFSC), tem como missão a pesquisa e o desenvolvimento de trabalhos e aplicações na área da Segurança da Informação, particularmente em Infraestruturas de Chaves Públicas e suas aplicações. O autor deste trabalho tem sido membro do laboratório desde o segundo semestre do ano de 2006 e já esteve envolvido em vários projetos.

Entre esses projetos, encontra-se o projeto João de Barro (ITI, 2008), mantido pelo Instituto Nacional de Tecnologia da Informação (ITI). Nesse projeto, o LabSEC foi responsável pelo desenvolvimento dos sistemas de gerenciamento de certificados digitais da autoridade certificadora raiz brasileira.

Durante o desenvolvimento desses projetos, diversas pesquisas foram realizadas sobre os melhores meios de se implementar uma aplicação de gestão de certificados digitais. Essas pesquisas resultaram em diversas propostas que aprimoraram a segurança e usabilidade das aplicações desenvolvidas. Este trabalho reúne os principais resultadas dessas pesquisas, de forma que o conhecimento produzido esteja registrado como um modelo para Sistemas Gerenciadores de Certificados Digitais preparado para a emissão distribuída e em larga escala de certificados digitais.

## 1.4 JUSTIFICATIVA

Embora existam modelos para a emissão de certificados digitais propostos na literatura, não há um modelo que possibilite a emissão de certificados digitais de forma distribuída e em larga escala. Pode parecer que se trata de um problema eminentemente de engenharia de sistemas usando artifícios para solucionar o problema, tal como a distribuição de cargas dos servidores das autoridades certificadoras. Entretanto, a característica única de controle da chave privada da autoridade certificadora faz com que tal abordagem não seja adequada para o problema posto. Também não se trata de um problema clássico de engenharia de software, ou seja, a inexistência de especificações e modelos de implementação de sistemas de autoridades certificadoras. Ainda não é claro para os pesquisadores, quais são os reais problemas relacionados ao ciclo de vida de chaves públicas. A falta de uma teoria que modele o problema tem levado a especificação de sistemas de ICPs que não atendem aos diversos requisitos de segurança e de desempenho. Grandes implementações de tais modelos tem sido alvo de críticas, tanto das entidades mantenedoras dos sistemas, quanto dos demais usuários dos serviços providos por ICPs. Este trabalho visa contribuir com um modelo genérico o suficiente para aten-

der a todos os requisitos de segurança inerentes a gestão do ciclo de vida de chaves criptográficas e certificados digitais. Tal modelo pode ser adotado nas situações de alta demanda de certificados digitais, tais como aquelas governamentais.

## 1.5 METODOLOGIA

Usou-se neste trabalho o método indutivo, no qual buscou-se a experiência que havia quanto aos sistemas de gerenciamento de certificados digitais. A partir deste conhecimento, buscou-se a generalização derivada de observações de casos de sistemas reais e de constatações particulares, relacionadas às dificuldades quanto a emissão de grande quantidade de certificados digitais.

Esta pesquisa pode ser classificada, do ponto de vista de sua natureza, como pesquisa aplicada. O objetivo foi gerar conhecimento para aplicações práticas dirigida à solução específica de como emitir certificados de forma distribuída em larga escada. Do ponto de vista da forma de abordagem ao problema, este trabalho é descritivo, pois os resultados foram analisados de forma indutiva. Do ponto de vista dos objetivos, trata-se de uma pesquisa exploratória, no sentido que proporciona uma maior familiaridade com o desafio em emitir grandes quantidades de certificados digitais. O trabalho envolveu um levantamento bibliográfico e também uma análise de vários sistemas de gerenciamento de certificados digitais proporcionados pelo laboratório onde a pesquisa foi realizada. Do ponto de vista dos procedimentos técnicos, trata-se de uma pesquisa bibliográfica, documental em alguns pontos onde havia pouca documentação, e experimental no sentido que foi especificado e implementado um protótipo de forma a avaliar o modelo proposto.

## 1.6 ORGANIZAÇÃO DO TRABALHO

Este trabalho é organizado em 6 capítulos, além da Introdução, conforme descrito a seguir.

- O **Capítulo 2** apresenta uma revisão bibliográfica sobre Infraestruturas de Chaves Públicas. Nele são apresentados os conceitos básicos de ICP;
- O **Capítulo 3** apresenta uma análise sobre os padrões da ICP *X.509* e as aplicações que os implementam. O objetivo dessa análise é construir uma modelo detalhado, *estendido do modelo X.509*, para Sistema Gerenciador de Certificados Digitais;

- O **Capítulo 4** apresenta o novo modelo de gerenciamento de certificados digitais baseado no modelo *X.509*.
- O **Capítulo 5** apresenta a especificação de um modelo de implementação para Sistemas Gerenciadores de Certificados Digitais adequados à emissão em larga escala de certificados digitais;
- O **Capítulo 6** apresenta uma discussão sobre as vantagens e desvantagens do modelo proposto;
- O **Capítulo 7** apresenta as considerações finais e trabalhos futuros.

# 2 INFRAESTRUTURA DE CHAVES PÚBLICAS

## 2.1 INTRODUÇÃO

Basicamente, existem duas categorias de criptografia: a criptografia simétrica e a criptografia assimétrica. Na criptografia simétrica, a chave criptográfica utilizada para cifrar uma mensagem é também utilizada para decifrar essa mensagem cifrada. Na criptografia assimétrica são utilizados pares de chaves, de tal forma que uma mensagem cifrada com uma dessas chaves só pode ser decifrada pela outra chave do par.

**Definição 2 (Chave Simétrica)** Uma chave simétrica, em conjunto com seus algoritmos de cifração e decifração de dados, pode ser definida como uma função $f_s$, tal que $f_s = f_s^{-1}$. Ou seja, a função $f_s$ é usada tanto para cifrar, como para decifrar o que cifrou.

**Definição 3 (Chaves Assimétricas)** Um par de chaves assimétricas, em conjunto com seus algoritmos de cifração e decifração, podem ser definidos como uma tupla de funções $K = (f_c, f_d)$, onde $f_d = f_c^{-1}$, e $f_d$ não pode ser deduzida a partir de $f_c$.

Neste capítulo é apresentada a fundamentação teórica para Infraestruturas de Chaves Públicas. Para isso, são apresentados os conceitos básicos da criptografia de chave pública, para então serem apresentados os componentes que formam a ICP. A primeira seção apresenta o histórico da criptografia de chaves pública, do seu surgimento até a conceitualização do certificado digital. As próximas seções apresentam os componentes que formam a Infraestrutura de Chaves Públicas *X.509*, modelo mais aceito pela comunidade mundial.

## 2.2 HISTÓRICO DA CRIPTOGRAFIA DE CHAVE PÚBLICA

As Infraestruturas de Chaves Públicas têm como base a criptografia de chave pública, originada do protocolo proposto por Diffie e Helman (DIFFIE; HELLMAN, 1976) para troca de chaves criptográficas. O protocolo proposto por eles permite que, através da troca de dados não sensíveis (públicos), duas entidades possam gerar uma mesma chave criptográfica convencional.

O protocolo está ilustrado na figura 1, onde Alice e Bob são as enti-

dades que desejam realizar a troca de chaves. As caixas ao lado deles listam as operações realizadas por eles, enquanto as setas valoradas representam as trocas de mensagens. Todo protocolo acontece através de um canal de comunicação aberto (sem criptografia), ou seja, os dados enviados nas mensagens podem ser considerados públicos. As operações e mensagens estão numeradas apenas para facilitar a compreensão do fluxo do protocolo e não representam uma restrição da sequência de passos.

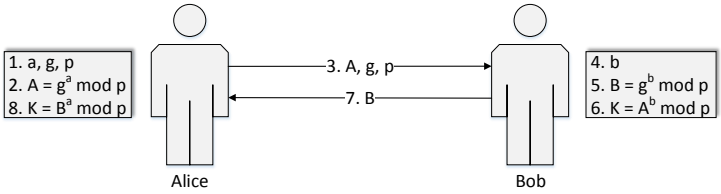


Figura 1 – Protocolo Diffie-Helman.

O primeiro passo do protocolo ilustrado na Figura 1 é a geração dos números $p$, $g$ e $a$, sendo $p$ um número primo, $g$ uma raiz primitiva módulo $p$, e $a \in \mathbb{N}$. O segundo passo consiste no cálculo de um número $A$, sendo $A = g^a \bmod p$. O terceiro passo é Alice enviar $p$, $g$ e $A$, para Bob. Então, como quarto, quinto e sexto passos, Bob gera um número $b$, tal que $b \in \mathbb{N}$; calcula $B$, sendo $B = g^b \bmod p$; e calcula $K$, sendo $K = A^b \bmod p$. Então, como sétimo passo, Bob envia para Alice o valor calculado $B$. Finalmente, como o oitavo e último passo, Alice calcula o número $K$, sendo $K = B^a \bmod p$. Os valores $K$ calculado por Alice e Bob são iguais, pois, na aritmética modular, o valor resultante de $(g^b \bmod p)^a \bmod p$ é sempre igual a $(g^a \bmod p)^b \bmod p$. Como apenas os número $p$, $g$, $A$ e $B$ são públicos e não é conhecido um algoritmo eficaz para calcular $\log_g(B)$ e $\log_g(A)$, então não é possível recuperar o número $K$ através dos dados públicos do protocolo.

Ronald Rivest, Adi Shamir e Leonard Adleman, embasados na publicação de Diffie e Helman, propuseram em 1978 um novo algoritmo de cifração e decifração de dados (RIVEST; SHAMIR; ADLEMAN, 1978). O algoritmo, chamado RSA, introduziu o conceito de par de chaves assimétricas, no qual uma mensagem cifrada por uma das chaves só poderia ser decifrada com a outra chave do par. Através dessa propriedade, foram propostas duas aplicações para as chaves assimétricas, o sigilo e a autenticação. Essas aplicações são ilustradas pelas Figuras 2 e 3, respectivamente.

Para gerar um par de chaves RSA é preciso computar o número inteiro $n$ através da multiplicação de dois número primos grandes $p$ e $q$, tal que $n =$

$p.q$. Então, deve-se escolhe um número inteiro grande $d$, tal que $d$ seja um primo relativo de $(p-1).(q-1)$. Por fim, se calcula o número inteiro $e$ de forma que ele seja o inverso multiplicativo de $d$ módulo $(p-1).(q-1)$, ou seja, $e.d \equiv 1(\bmod (p-1).(q-1))$ (RIVEST; SHAMIR; ADLEMAN, 1978).

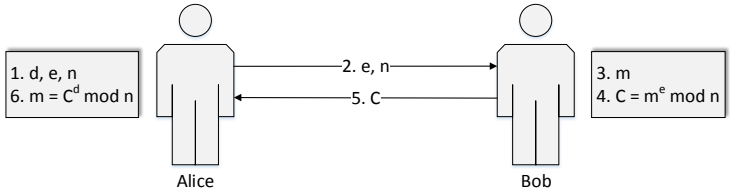


Figura 2 – Sigilo RSA.

O processo utilizados para a aplicação de sigilo do RSA consiste nos seguintes passos. Após gerar o par de chaves assimétricas, Alice envia para Bob uma das chaves do seu par de chaves. Essa chave pode ser enviada via um canal aberto, portanto, ela é chamada de chave pública. Então, Bob, utiliza a chave pública para cifrar sua mensagem, através da seguinte operação: $C = m^e \bmod n$. Por fim, Bob envia a mensagem cifrada $C$ para Alice, que a decifra através da seguinte operação: $m = C^d \bmod n$. Como apenas Alice tem acesso ao número $d$, ou a chave privada, então apenas ela será capaz de decifrar a mensagem $C$.

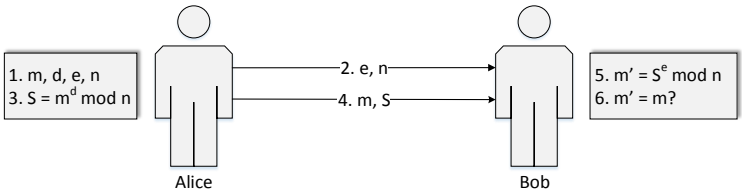


Figura 3 – Autenticação RSA.

O processo utilizado para a aplicação de autenticação do RSA se inicia da mesma forma que o processo de sigilo. Após gerar o par de chaves, Alice envia para Bob uma delas, a chamada chave pública. Então, Alice cifra sua mensagem com a chave que manteve secreta, a chave privada, através da seguinte operação: $S = m^d \bmod n$. Alice envia $S$ e $m$ para Bob, que decifra $S$ com a chave pública de Alice, através da seguinte operação: $m = S^e \bmod n$.

Então, Bob verifica se o valor de *m* calculado é igual ao valor de *m* enviado por Alice. Sendo iguais, Bob sabe que Alice enviou a mensagem, pois apenas ela possui a chave privada capaz de gerar *S*.

Por se assemelhar à ideia da assinatura em papel, que é usada para autenticar uma pessoa, o processo de cifrar algo com uma chave privada se tornou conhecido como assinatura digital. Existem diversos trabalho relacionados à criação e validação de assinaturas digitais, que especificam algoritmos seguros e padronizações. Porém, por não ser o foco deste trabalho, os processos de criação e validação de assinatura digital serão representados apenas pela cifração com a chave privada e decifração com a chave pública.

Na criptografia convencional, ou criptografia simétrica, a chave criptográfica tem que ser tão secreta quanto a mensagem que se deseja cifrar, pois a mesma chave é usada nos processos de cifração e decifração. Dessa forma, o usuário da chave simétrica precisa ter certeza de que a chave que esta usando só foi entregue para as pessoas que podem ler àquela mensagem. Ou seja, a troca de chaves precisa ocorrer através de um canal protegido e autenticado.

Na criptografia assimétrica, o controle da chave de cifração é mais simples, pois essa chave é pública. Mesmo que um atacante tenha acesso à chave (e provavelmente terá), ele não poderá fazer nada com ela, pois apenas a chave privada poderá decifrar as mensagem cifradas pela chave pública. Porém, na criptografia assimétrica, o usuário da chave pública ainda precisa ter conhecimento de quem é o custodiante da chave privada correspondente, ou seja, ele precisa obter a chave pública através de um canal autenticado. Se não, um atacante pode realizar um ataque de personificação, como o ataque do homem do meio (Figura 4).

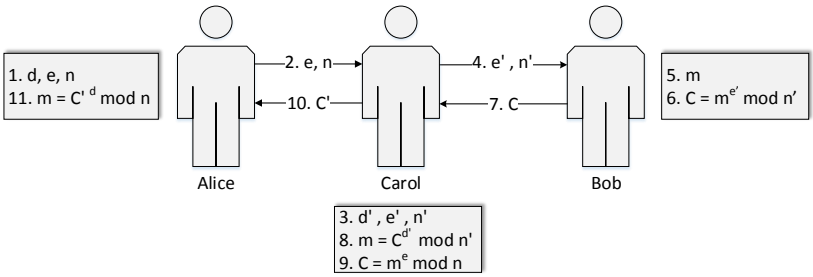


Figura 4 – Ataque do Homem do Meio.

A criação de um canal autenticado é tão complexa quanto a criação de um canal sigiloso. Porém, diferente da criptografia simétrica, as chave públi-

cas podem ser armazenadas em um diretório público, que se responsabiliza pela identificação dos custodiantes das chaves privadas correspondentes. O uso de um diretório público permite que, com apenas um canal autenticado, seja possível obter a chave pública de qualquer pessoa que tenha sua chave cadastrada nele.

A Figura 5 ilustra um exemplo de como um diretório público pode funcionar. Ao invés de Alice enviar sua chave pública diretamente para Bob, ela a cadastra no Diretório Público, através de um canal autenticado que garanta a autenticidade de Alice. Então, através de um canal autenticado que garanta a autenticidade do Diretório Público, Bob recupera a chave pública de Alice. Dessa forma, Bob tem certeza que a chave pública que possui é de Alice.

Através do Diretório Público, ao invés de Alice ter que construir um canal autenticado para cada pessoa que enviar sua chave pública, ela apenas precisa gerar um canal autenticado com o Diretório Público. Da mesma forma, as pessoas que desejam ter acesso à diversas chaves públicas precisam criar apenas um canal autenticado com o Diretório Público. A vantagem fica visível quando grupos grandes de pessoas desejam se comunicar umas com as outras, como mostrado na Figura 6.

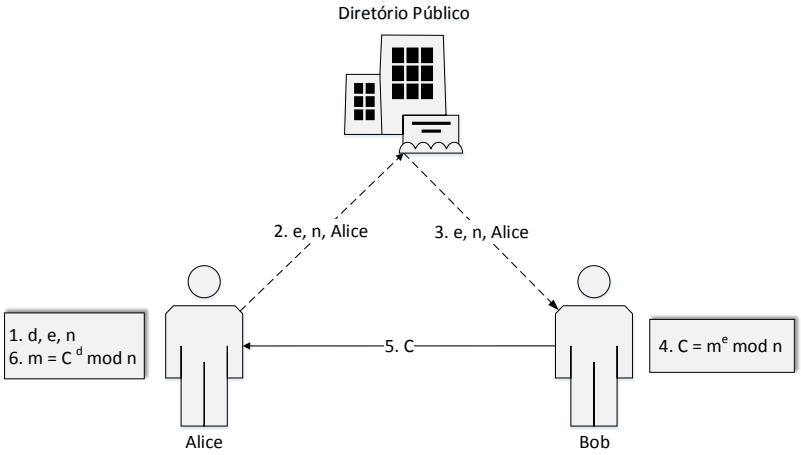


Figura 5 – Diretório Público.

Apesar do uso do Diretório Público ser uma grande evolução no uso da criptografia no meio digital, ele ainda é suscetível a um ataque de negação de serviço. Como o diretório precisa ser consultado para se estabelecer

uma conexão segura com seus usuários, basta impedir a comunicação com o diretório para impedir a formação de novos canais seguros, durante o tempo de duração do ataque. Como solução para esse problema, Kohnfelder propôs, pela primeira vez, o uso de certificados digitais.

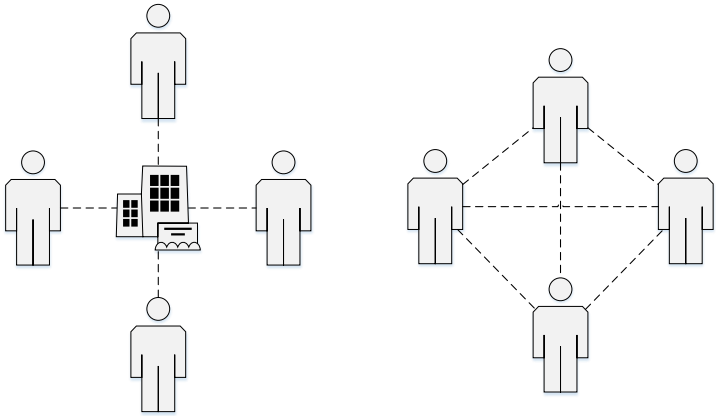

Figura 6 – Centralização.

## 2.3 CERTIFICADO DIGITAL

O conceito de certificado digital foi apresentado por Khonfelder como uma forma de facilitação do gerenciamento de chaves públicas. Sua proposta foi criar uma estrutura de dados contendo uma chave pública, o nome em texto plano do custodiante da chave, e um autenticador, responsável pela autenticidade do vínculo da chave pública com o nome de seu custodiante (KOHNFELDER, 1978).

Atualmente, os padrões relacionados à estrutura de dados do certificado digital, seu preenchimento e interpretação, são definidos pela RFC-5280 (COOPER et al., 2008). Como proposto por Khonfelder, a estrutura X.509 associa uma chave pública ao nome de seu custodiante, através de uma assinatura digital realizada sobre essas informações. A estrutura é codificada em ASN.1[1](International Telecommunication Union, 2002), conforme ilustra a figura 7.

O campo *tbsCertificate* é um estrutura que contém todas informações

[1]ASN.1 - `http://www.itu.int/ITU-T/studygroups/com17/languages/`

```
Certificate ::= SEQUENCE {
    tbsCertificate          TBSCertificate,
    signatureAlgorithm      AlgorithmIdentifier,
    signatureValue          BIT STRING }
```

Figura 7 – Estrutura ASN.1 de um Certificado Digital X.509.

que devem ser assinadas. Entre elas, encontram-se: a chave pública, no campo *subjectPublicKeyInfo*; e o nome do custodiante da chave, no campo *subject*. Os campos *SignatureAlgorithm* e *signatureValue* são utilizados para guardar o algoritmo de assinatura e o valor da assinatura realizada sobre o campo *tbsCertificate*. Esses campos são equivalentes a tripla proposta por Khonfelder: informações da chave pública, nome do custodiante da chave e o autenticador.

**Definição 4 (Certificado Digital)** O certificado digital pode ser representado por uma tripla $C = (Au, f_c, U)$, onde $Au$ um autenticador, gerado por uma função $f'_d$, que vincula a função de cifragem $f_c$ às informações do detentor de $f_c^{-1}$, representadas por $U$ (KOHNFELDER, 1978). Neste trabalho definimos $U$ como uma tupla contendo as informações do campo *tbsCertificate*, definido pelo padrão *X.509* (COOPER et al., 2008).

Com essa estrutura é possível identificar o dono da chave, sem ser necessário consultar um diretório público de chaves, como ilustrado pela Figura 8. O processo se assemelha ao processo descrito na Figura 5, porém, ao invés de Alice cadastrar sua chave no Diretório Público, Alice solicita um certificado digital. Para emitir o certificado de Alice, o Diretório Público concatena a chave pública de Alice com o seu nome, para então assinar essa informação com sua chave privada. Com a chave pública de Alice, seu nome e a assinatura realizada pelo Diretório Público, o certificado é construído e enviado para Alice. A partir daí, o certificado pode ser distribuído de inúmeras formas, pois a autenticidade do vínculo entre Alice e sua chave pública é garantida pela assinatura do Diretório Público.

Para se comunicar com Alice, Bob pode recuperar o certificado digital de Alice de qualquer uma das fontes disponíveis. Então, através de um canal autenticado, ele obtém a chave pública do Diretório Público e a usa para verificar a autenticidade do certificado digital de Alice. Porém, após obter a chave pública do Diretório Público, Bob pode armazená-la para não ter que consultá-lo novamente quando for verificar um outro certificado digital.

Em seu trabalho, Khonfelder também aborda o problema relacionado ao comprometimento da chave privada relacionada ao certificado digital. Nes-

ses casos, ele afirma que devem existir mecanismos de invalidação do certificado digital, para que a chave comprometida não possa ser usada por um atacante. Esses mecanismos são descritos na próxima seção.

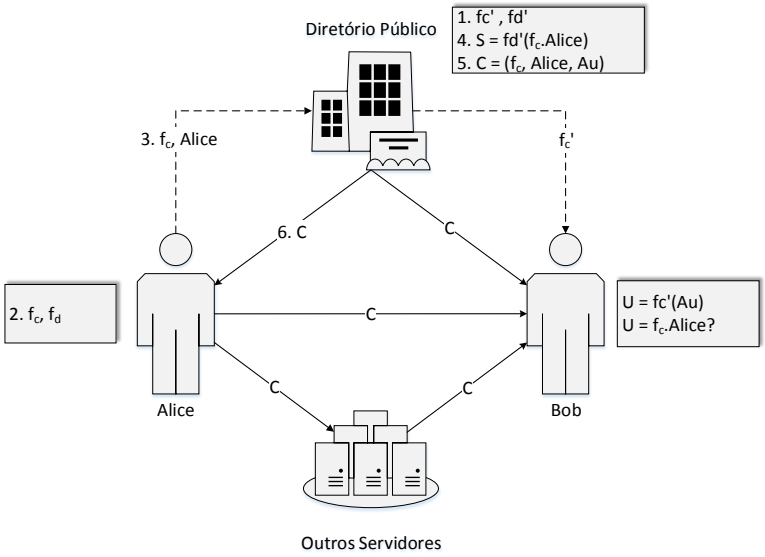


Figura 8 – Emissão e Distribuição do Certificado Digital.

## 2.4 LISTA DE CERTIFICADOS REVOGADOS

O comprometimento de uma chave privada é um dos maiores problemas para sistemas baseados em criptografia de chave pública (KOHNFELDER, 1978). Uma vez que um atacante tem acesso a uma chave privada, todas mensagens sigilosas, criadas antes do comprometimento da chave, poderão ser lidas pelo atacante. Além disso, no intervalo de tempo entre o comprometimento da chave e a descoberta desse comprometimento, o atacante poderá assinar mensagens em nome do custodiante da chave. Atualmente não existem soluções ótimas para esses problemas, mas algumas medidas podem ser tomadas para reduzir os prejuízos.

Khonfelder propôs em seu trabalho três soluções para a identificação de chaves comprometidas. A primeira delas é a criação de uma Lista de Cer-

tificados Revogados (LCR), onde estariam identificados os certificados cujas chaves foram comprometidas. Nessa solução, todos interessados devem atualizar uma lista própria, conforme recebem informações sobre o comprometimento de chaves. A segunda solução é definir datas de validade para os certificados digitais, forçando a atualização das suas chaves de tempos em tempos. A terceira solução é realizar consultas ao Diretório Público, sobre a validade do certificado digital, permitindo o usuário realizar uma dupla checagem da validade do certificado digital (KOHNFELDER, 1978). Essas duas últimas propostas, conforme Khonfelder, iam contra o objetivo do uso de certificados digitais, que é reduzir o número de consultas ao Diretório Público.

As três soluções acabaram sendo usadas em uma solução híbrida, proposta pela ICP *X.509*. Nela, são usadas as datas de validade para os certificados digitais, uma Lista de Certificados Revogados (LCR) e a consulta à uma entidade responsável sobre a validade do certificado digital. A validade do certificado digital é especificada em sua estrutura, como mostrado na Seção anterior. A consulta sobre a validade do certificado é padronizada por um protocolo especificado pela RFC-6960 (*Online Certificate Status Protocol - OCSP*) (SANTESSON et al., 2010).

A atualização da LCR é responsabilidade de uma terceira parte confiável, como o Diretório Público, que deve assinar o conteúdo da lista, para criar provas da sua autenticidade. As especificações das estruturas e procedimentos envolvidos com a LCR são padronizados pela RFC-5280 (COOPER et al., 2008), da mesma forma que o certificado digital. Sua estrutura básica é apresentada logo abaixo. O campo *tbsCertList* contém a lista de certificados revogados e outras informações relativas a LCR, como data de emissão e data prevista para a próxima emissão de LCR. O campo *signatureAlgorithm* e *signatureValue* são utilizados para armazenar as informações sobre o algoritmo de assinatura e o valor da assinatura realizada sobre o campo *tbsCertList*, respectivamente.

**Definição 5 (Lista de Certificados Revogados)** Uma lista de certificados revogados pode ser representada por uma tupla $L = (Au, lu, nu, R)$, onde $A_u$ é um autenticador gerado por uma função $f'_d$ sobre um estrutura de dados contendo: $lu$, como a data de emissão da LCR; $nu$, como próxima atualização da LCR; e $R$, como um conjunto de certificados revogados.

A figura 9 apresenta a estrutura X.509 de uma lista de certificados digitais em ASN.1.

Através desse modelo, uma vez que uma chave privada é comprometida, o dono da chave deve informar o ocorrido à entidade responsável por emitir a LCR. Essa entidade deve atualizar a Lista de Certificados Revogados

```
CertificateList  ::=  SEQUENCE  {
    tbsCertList          TBSCertList,
    signatureAlgorithm   AlgorithmIdentifier,
    signatureValue       BIT STRING }
```

Figura 9 – Estrutura X.509 da Lista de Certificados Revogados em ASN.1.

com o identificador do certificado revogado, assiná-la e publicá-la. Então, para validar um certificado digital, os interessados devem recuperar a LCR mais atual e verificar se o certificado sendo verificado não consta nela.

## 2.5 MODELOS DE CONFIANÇA

Um modelo de confiança define a organização estrutural e lógica das relações de confiança estabelecidas entre um conjunto de entidades. Por exemplo, quando Bob solicita a chave pública de Alice ao Diretório Público, ele confia no Diretório Público para retornar a chave pública correta de Alice. De forma semelhante, quando Alice cadastra sua chave no Diretório Público, ela deve se autenticar, para que o Diretório Público confie na relação entre ela e sua chave pública. Essas relações de confiança podem ser expressas na forma de um grafo orientado, onde os nodos representam as entidades do modelo e as arestas as relações de confiança.

A Figura 10 ilustra a relação de confiança entre Bob, Alice e o Diretório Público. As setas saem da entidade que confia para a entidade confiada. A definição semântica de confiança varia, dependendo do contexto da relação. Essa variação é ilustrada pelo estilo da aresta entre as entidades. Nesse caso, a seta preenchida representa a confiança de Bob sobre o Diretório Público fornecer relações entre entidades e chaves de forma confiável, enquanto a seta tracejada representa a confiança do Diretório Público na relação entre Alice e sua chave pública. Por fim, a seta pontilhada representa a confiança derivada de uma sequência concatenada de relações de confiança compatíveis. Ou seja, Bob confia no Diretório Público para fornecer relações entre entidades e chaves públicas e o Diretório confia na relação entre Alice e sua chave Pública, portanto a confiança de Bob transita do Diretório Público para a relação entre Alice e sua chave pública.

Existem diversos trabalhos que exploram as especificações e interpretações dos modelos de confiança. Porém, por não ser o foco desse trabalho, todas relações de confiança são tratadas de forma simplificada, sem explicitar suas diferenças semânticas. Para isso, deve-se considerar que todas as relações de confiança apresentadas a partir de agora são compatíveis e transi-

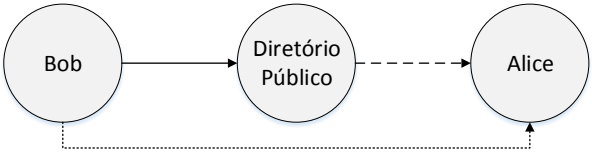


Figura 10 – Modelo de Confiança.

táveis.

Utilizando essa notação, é possível abstrair o certificado digital como uma relação de confiança entre a entidade que assinou o certificado e a entidade responsável pelo certificado. Seguindo essa lógica, é possível definir diversas organizações estruturais para a Infraestrutura de Chaves Públicas. A Figura 11 ilustra uma das organizações possíveis, conhecida como organização em rede. Uma implementação muito conhecida desse modelo é a infraestrutura de chaves pública PGP (*Preety Good Privacy*) (GARFINKEL, 1995). Esse formato de organização se assemelha a organização natural de confiança entre pessoas, que não possui um ordem restrita de hierarquia.

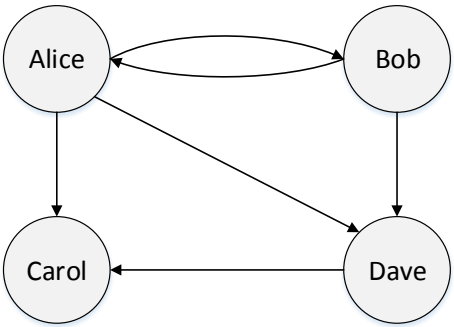


Figura 11 – Exemplo de Organização em Rede.

O problema com o modelo de confiança em rede é a definição de quando ou não confiar em um certificado digital. Por exemplo, na rede ilustrada pela Figura 11, Bob confia diretamente em Alice e Dave, mas não confia em Carol. Porém, Alice e Dave confiam em Carol. Portanto, Bob deveria confiar em Carol? Para responder essa pergunta, é preciso que Bob estabeleça

uma métrica de confiança. Por exemplo, ele pode estabelecer que as relações de confiança mais próximas são mais confiáveis, enquanto relações derivadas são menos confiáveis. Nesse caso, Carol seria menos confiável que Alice e Dave. Outra opção é definir que as entidades que possuem o maior número de assinaturas são mais confiáveis, nesse caso Carol seria até mais confiável do que Alice.

Em determinados contextos, como em uma ICP governamental, a subjetividade das relações de confiança não são toleradas. Portanto, o modelo de confiança em rede não é considerado uma boa solução para esses contextos. Para se remover a subjetividade do modelo de confiança, é necessário estabelecer entidades especializadas em emitir certificados digitais, que se responsabilizam, legalmente, pela autenticidade das informações contidas neles. Essas entidades são chamadas de Autoridades Certificadoras (AC).

### 2.5.1 Autoridade Certificadora

Conforme Housley, uma Autoridade Certificadora é definida por um conjunto de *hardware*, *software* e pessoas, que trabalham em conjunto para fornecer as funções de emissão de certificado digital, revogação de certificado digital, emissão de LCR e publicação dessas informações (HOUSLEY; POLK, 2001).

A autoridade certificadora é identificada através de um certificado digital, com extensões que identificam seu estado de autoridade. Esse certificado pode ser autoassinado e publicado através de um canal autenticado, para que qualquer pessoa, ou máquina, possa fazer a verificação de sua validade. A instituição que opera a autoridade certificadora é responsável pela segurança de sua chave privada, pois, se comprometida, todos certificados emitidos por ela são invalidados. Para isso, diversas regras de segurança devem ser especificadas e seguidas pela AC.

As autoridades certificadoras, então, são as únicas entidades capazes de emitir certificados digitais válidos, centralizando a confiança dos usuários da ICP. Porém, as autoridades certificadoras podem delegar suas funções para outras autoridades certificadoras. Isso é feito através da emissão de um certificado de uma AC para outra. Essa delegação costuma ser organizada de forma hierárquica, como ilustrado pela Figura 12.

Nessa grafo, os nodos representam Autoridades Certificadoras (círculos) e Usuários Finais (retângulos), enquanto as setas representam os certificados digitais, sendo que a seta sai da AC emissora e chega na AC ou Usuário responsável pelo certificado. No topo da hierarquia encontra-se a Autoridade Certificadora Raiz, com um certificado autoassinado. Essa AC é o ponto cen-

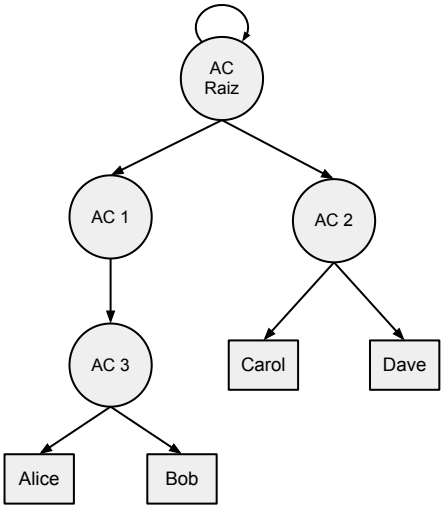


Figura 12 – Exemplo de Organização Hierárquica de ICP

tral de confiança, no qual todos usuários da ICP devem confiar (ela também é conhecida como âncora de confiança). Abaixo da AC Raiz existem as ACs Subordinadas de primeiro nível, com certificados emitidos pela AC Raiz. As ACs subordinadas podem ser intermediárias ou finais, sendo que as intermediárias emitem certificados para outras ACs, enquanto as finais emitem certificados para usuários finais.

### 2.5.2 Autoridade de Registro

As autoridades de registro (AR) surgiram para assumir as atividades de identificação e autenticação dos usuários finais, necessária quando a emissão de um certificado é solicitada a uma AC. Por ser a entidade que se relaciona diretamente com o usuário, a AR é considerada como a interface de comunicação entre o requerente de certificado e a AC. Da mesma forma que a AC, a AR é definida como um conjunto de *hardware*, *software* e pessoas, que trabalham em conjunto para realizar as funções da AR (HOUSLEY; POLK, 2001). As funções da AR estão relacionadas diretamente com as regras exigidas pela AC para identificar seus requerentes. A autenticação pode ir de uma prova de posse de email até a validação presencial de documentos oficiais que identificam o requerente de certificado.

A autoridade de registro também pode ser identificada por um certificado digital, porém, esse certificado deve ser semelhante a um certificado de usuário final. Dessa forma, uma AC que deseja usar os serviços de uma AR deve adicionar o certificado da AR em uma lista de ARs confiáveis. Assim, toda vez que a AR enviar uma mensagem para a AC, a AR a assina com sua chave privada, permitindo que a AC verifique se a mensagem é proveniente de uma AR confiável. Um Usuário Final, que deseja obter um certificado da AC, deve se comunicar com a AR relacionada a ela. Assim, a AR poderá validar a requisição de certificado do usuário final e enviar uma mensagem, assinada, para a AC emitir o certificado digital.

## 2.6 CONCLUSÃO

Neste capítulo foram apresentados os fundamentos da infraestrutura de chaves públicas. Essa fundamentação é importante, pois define terminologias que serão utilizadas ao longo de todo trabalho. Além disso, o capítulo mostra os benefícios trazidos pelo uso da criptografia assimétrica, em comparação com os métodos tradicionais de criptografia, como a distribuição de chaves e o conceito de assinatura digital.

A infraestrutura de chaves públicas, por suas vez, é apresentada como uma solução estrutural e lógica de como as chaves públicas, utilizadas na criptografia assimétrica, devem ser distribuídas e utilizadas. Este trabalho se foca na infraestrutura de chaves públicas X.509, sendo essa solução a mais aceita e utilizada internacionalmente.

Todas as entidades e artefatos que formam a ICP X.509 interagem para controlar o ciclo de vida dos certificados digitais. Porém, neste capítulos, essas entidades e artefatos são apresentados superficialmente, nas suas formas conceituais. Quando implementada, a ICP X.509 se mostra mais complexa do que aparenta na sua forma conceitual. No próximo capítulo, essa complexidade é exposta, através do detalhamento do ciclo de vida dos certificados digitais e do destrinchamento das entidades que forma a ICP X509, na forma de protocolos e sistemas gerenciadores de certificados digitais.

# 3 CICLO DE VIDA DOS CERTIFICADOS DIGITAIS

## 3.1 INTRODUÇÃO

Para ser possível propor um modelo geral para a emissão em larga escala de certificados digitais é preciso conhecer o ciclio de vida dos certificados digitais e, neste ciclo, as tarefas associadas à emissão propriamente dita dos certificados. A Seção 3.2 apresenta os principais estados em que um certificado pode estar. A Seção 3.3 descreve o modelo básico proposto pelo grupo PKIX da IETF. Os SGCs em geral, seguem as recomendações deste modelo. A Seção 3.4 apresenta alguns SGCs reais do quais teve-se contato durante o desenvolvimento deste trabalho. A Seção 3.5 analisa esses sistemas à luz das recomendações e normais existentes. A Seção 3.6 conclui o capítulo.

## 3.2 CICLO DE VIDA BÁSICO

O ciclo de vida dos certificados digitais pode ser definido em três estados principais: *emitido*, *revogado* e *expirado*. O estado *emitido* é estabelecido logo após a criação do certificado digital, ou seja, após ele ser emitido por uma Autoridade Certificadora. O estado *revogado* é estabelecido quando uma Autoridade Certificadora emite uma Lista de Certificados Revogados que inclui o certificado digital. Por fim, o estado *expirado* é estabelecido quando a data de validade do certificado é ultrapassada. O ciclo entre esses estados é ilustrado pela Figura 13.

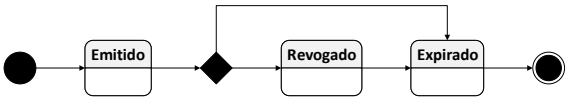


Figura 13 – Ciclo de Vida Simplificado do Certificado Digital

As entidades responsáveis por gerenciar o ciclo de vida dos certificados digitais são as Autoridades Certificadoras. Porém, Autoridades de Registro também podem fazer parte desse gerenciamento. Elas são responsáveis por interfacear a comunicação entre a AC e o Titular do Certificado, recebendo as solicitações de emissão de certificados digitais para, então, aprová-

las ou rejeitá-las. Esse procedimento inclui mais três novos estados para o ciclo de vida do certificado digital, definindo as etapas realizadas antes da emissão do próprio certificado. Esses estados e suas relações no ciclo de vida dos certificados digitais são apresentados na Figura 14.

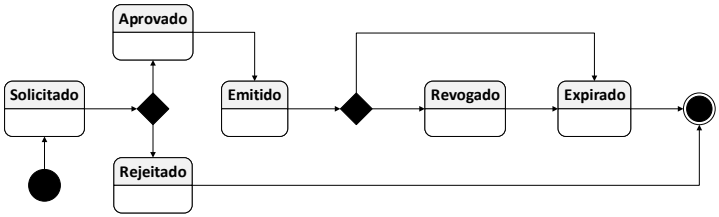


Figura 14 – Ciclo de Vida do Certificado Digital

Outro ponto importante a se destacar é que a revogação do certificado não ocorre em apenas um passo. Antes de um certificado ser revogado, sua revogação deve ser solicitada à AR e aprovada. Quando aprovada, o certificado digital será marcado para revogação, sendo a revogação efetivada apenas na próxima emissão de LCR. Esses processos envolvem mais quatro estados para o ciclo de vida do certificado digital. Porém, para simplificar o diagrama de máquina de estados, esses estados são apresentado em uma máquina separada (Figura 15), que tem seu estado inicial entre os estados *emitido* e *revogado* da máquina ilustrada pela Figura 14. Além disso, todos esses novos estados podem transitar para o estado *expirado*, caso a data de validade do certificado seja ultrapassada.

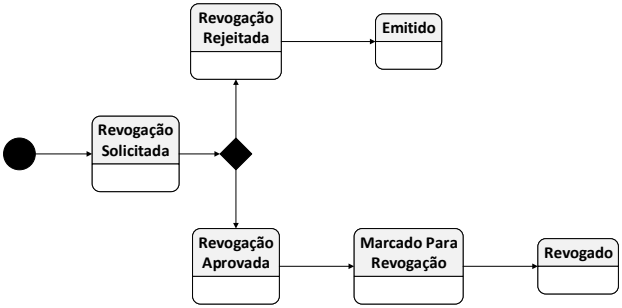


Figura 15 – Revogação no Ciclo de Vida do Certificado Digital

Para gerenciar o ciclo de vida dos certificados digitais, ACs e ARs

utilizam aplicações, máquinas e pessoas que constituem um Sistema Gerenciador de Certificados Digitais. De uma forma geral, esses sistemas permitem emitir certificados digitais, revogá-los e emitir listas de certificados revogados. Atualmente, existem diversos trabalhos na literatura que especificam padrões para serem seguidos por esses sistemas. Neste Capítulo, são apresentadas as análises realizadas sobre as especificações da ICP *X.509* e seu modelo de Gerenciamento de Certificados Digitais, em comparação com implementações existentes desse modelo. Essa análise tem por objetivo destacar as funções não especificadas pela ICP *X.509* que costumam ser implementadas pelas implementações de SGC.

## 3.3 O MODELO DE GERÊNCIA DE CERTIFICADOS DA ICP *X.509*

A Infraestrutura de Chaves Públicas *X.509* possui diversos documentos que especificam estruturas de dados, protocolos, algoritmos, procedimento e padrões de interpretação de dados, para as entidades que a constituem. Esses documentos são escritos no formato de RFCs (Request For Comments) e mantidas pelo grupo de trabalho *pkix* (IETF, 2013) da IETF (Internet Engineering Task Force). Dentre eles, destacam-se:

- **RFC-5280:** A RFC-5280 é responsável pela especificação das estruturas e semânticas dos dados que compõem os Certificados Digitais e Listas de Certificados Revogados. Ela também especifica um algoritmo para validação de Certificados Digitais *X.509* (COOPER et al., 2008).

- **RFC-4211:** A RFC-4211 é responsável pela especificação das estruturas e semânticas de dados que compõem uma requisição de certificado digital. Ela foi desenvolvida para ser usada em conjunto com um protocolo de gerenciamento de certificados digitais (SCHAAD, 2005).

- **RFC-4210:** A RFC-4210 especifica o protocolo CMP (Certificate Managment Protocol) para gerenciamento de certificados digitais. Ele utiliza requisições de certificados no formato proposto pela RFC-4211; e especifica as trocas de mensagens, feitas pelas entidades da ICP, para realizar o gerenciamento de certificados digitais (ADAMS et al., 2005).

- **RFC-5272:** A RFC-5272, da mesma forma que a RFC-4210, especifica um protocolo de gerenciamento de certificados digitais, conhecido por CMC (Certificate Managment over CMS). Esse protocolo é desenvolvido sobre o padrão CMS (Cryptographic Message Syntax), e é compatível com requisições de certificados no formato *PKCS#10*. Ou-

tros aspectos do protocolo são especificados pelas RFCs 5273 e 5274 (SCHAAD; MYERS, 2008a, 2008b, 2008c).

- **RFC-6960:** A RFC-6960 especifica o protocolo OCSP (Online Certificate Status Protocol), utilizado para realizar consultas online ao estado de validade de certificados digitais. Esse protocolo tem por objetivo prover uma outra via, além da LCR, para validar certificados digitais (SANTESSON et al., 2010).

- **RFC-2986:** Apesar de não fazer parte das RFCs escritas pelo grupo de trabalho *pkix*, A RFC-2986 é amplamente utilizada pela comunidade como padronização das estruturas e semânticas dos dados de requisições de certificados digitais. Mais conhecida pelo nome PKCS#10, esse padrão foi desenvolvido pela *RSA Laboratories* e faz parte do conjunto de padrões para criptografia de chave pública, do inglês, Public-Key Cryptography Standards (PKCS) (NYSTROM; KALISKI, 2000).

Essas RFCs especificam padrões que devem ser implementados por todos Sistemas Gerenciadores de Certificados digitais que desejam ser compatíveis com a ICP *X.509*. Porém, as RFCs 4210 e 5272 são as mais próximas de um modelo de SGC. Elas especificam as entidades de uma ICP, suas funções e relações, para estabelecer o processo de emissão e revogação de certificados digitais. O diagrama da Figura 16 ilustra esse modelo.

Nesse diagrama, são apresentadas três entidades: o Titular do Certificado, a Autoridade Certificadora e a Autoridade de Registro. Além disso, o diagrama apresenta um repositório para publicação de certificados digitais e LCRs, que pode ser acessado por todas entidades. As setas no modelo representam canais de comunicação, por onde as entidades trocam mensagens, para executar as funções da ICP. Entre as funções previstas, pela RFC-4210, destacam-se:

- **Inicialização/Certificação:** As funções de inicialização e certificação são semelhantes. Ambas iniciam com uma requisição de certificado feita por um requerente e finalizam com a emissão desse certificado. Porém, a função de inicialização serve como uma identificação do momento em que o requerente se identifica e autentica, perante a AC ou AR, pela primeira vez. Sendo que, nesse momento, a AC ou AR pode registrar as informações do requerente para reduzir a burocracia nas próximas requisições de certificado digital.

- **Revogação:** A função de revogação de certificados digitais consiste na solicitação para uma AC ou AR revogar um certificado digital emitido por ela. Essa função não resulta na revogação efetiva e imediata do

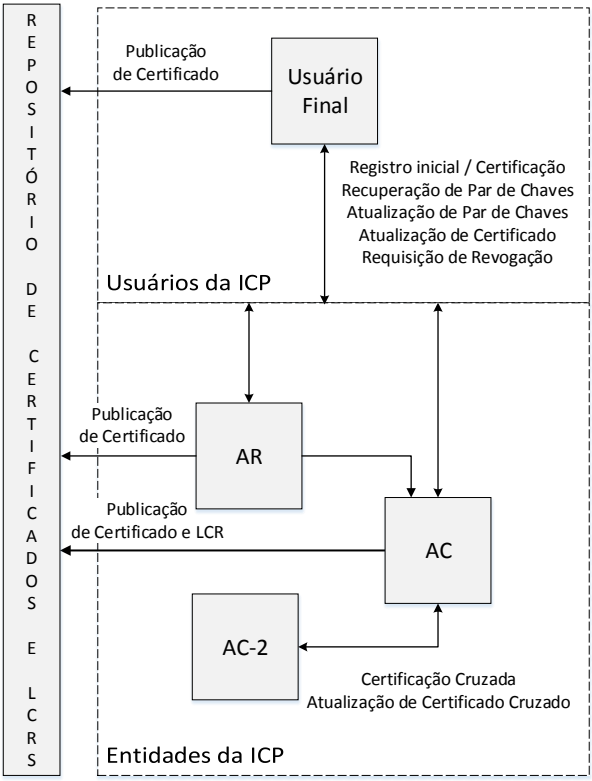


Figura 16 – Diagrama de Entidades da ICP *X.509*

certificado. O certificado é marcado como revogado e é inserido na próxima LCR emitida.

- **Atualização de Chave/Certificado:** A atualização de chaves consiste na emissão de um novo certificado digital com as mesmas informações de um certificado já emitido, mas com uma chave pública diferente. A atualização de certificado acontece após a validade de um certificado expirar, através da emissão de um novo certificado digital.

- **Recuperação de Chave:** A RFC 4210 oferece funções que dão suporte às Autoridades Certificadoras ou de Registro que fazem custódia de chave privada. Nesses casos, a função de recuperação de chave permite que um Titular do Certificado solicite sua chave privada à AC.

- **Certificação Cruzada:** A certificação cruzada consiste na emissão de um certificado digital para uma outra Autoridade Certificadora que já possui um certificado digital. Essa função é usada para na integração de ICPs.

- **Publicações:** A RFC 4210 define diversas funções de publicação de informação, entre elas: atualização da chave privada da AC, certificados emitidos, LCRs emitidas, certificado revogado, entre outras.

As funções especificadas pelas RFC 4210 são focadas nas necessidades do Titular do Certificado em relação à Autoridade Certificadora. Quando uma Autoridade de Registro é usada, ela é tratada como uma AC na visão do requerente, e tratada como um requerente na visão da AC. Ou seja, a AR é apenas um afunilamento para as requisições dos Titulares dos Certificados. Tanto, que a entidade AR é considerada opcional nesse modelo.

Apesar desse modelo especificar as estruturas e protocolos que devem ser implementados no nível da aplicação, ele ainda pode ser considerado uma macro-visão do Sistema Gerenciador de Certificados Digitais. As entidades desse modelo - AC, AR, Titular do Certificado - são conceituais, pois abstraem um conjunto muito mais complexo de aplicações, máquinas e pessoas. No processo de especificação de um SGC, pode-se encontrar um conjunto considerável de requisitos não previstos, ou não especificados, pelas RFCs. Para construir esse conjunto de requisitos, foram analisados Sistemas Gerenciadores de Certificados Digitais utilizados por entidades reais de AC e AR. Essa análise é apresentada nas próximas seções.

## 3.4 EXEMPLOS DE SGCS

Apesar da variedade de soluções para SGCs existentes no mercado, muitas instituições optam por desenvolver seus próprios sistemas, pois eles costumam estar envolvidos com informações sigilosas da instituição. Dessa forma, os próprios SGCs acabam se tornando sigilosos, dificultando qualquer estudo sobre eles. Porém, o Laboratório de Segurança em Computação pôde ter contato com alguns softwares desenvolvidos especificamente para algumas instituições, são eles:

- o Ywapa e Ywyra, desenvolvidos pelo Projeto João de Barro e usados pela AC Raiz e pelas ACs de Primeiro Nível da ICP-Brasil (ITI, 2008);

- o SGCI, desenvolvido pela ICPEdu (LabSEC, 2013), para ser usado em ACs e ARs do âmbito acadêmico.

Além desses SGCs, foram avaliados o *EJBCA Enterprise PKI CA* (PrimeKey, 2013) e alguns SGCs disponíveis no repositório de aplicações do Ubuntu: *Xca* (HOHNSTäDT, 2012), *Tinyca* (SM-Zone, 2006) e *Gnomint* (MARíN, 2006). A seguir, são apresentadas as principais características de cada um desses SGCs.

### 3.4.1 Ywapa e Ywyra

O Sistemas Gerenciadores de Certificados Digitais Ywapa e Ywyra (Raiz e Tronco, respectivamente, em Tupi-Guarani) foram desenvolvidos pelo programa João-de-Barro, que é mantido pelo Instituto Nacional de Tecnologia da Informação (ITI). Esse programa teve como objetivo desenvolver hardware e software nacionais, para serem usados pelas entidades da Infraestrutura de Chaves Públicas Brasileira (ICP-Brasil) (ITI, 2008). O Ywapa e o Ywyra, em particular, foram desenvolvidos para gerenciar os certificados das ACs Raízes e Intermediárias (offline) dessa ICP.

Efetivamente desenvolvidos pelo LabSEC, esses SGCs são implementados em C++ e funcionam em um ambiente completamente offline. Seus desenvolvimentos foram baseados em um conjunto de requisitos de software que estão de acordo com os níveis de segurança exigidos por Autoridades Certificadoras Governamentais. Entre esses requisitos, destacam-se: a compatibilidade com módulos de segurança criptográficos, o controle de acesso resistente a corrupção de usuários e o sistema de backup protegido contra leitura e escrita não autorizadas (WERLANG; MARTINS, 2010).

A compatibilidade com Módulos de Segurança Criptográficos é um requisito implementado por muitos Sistemas Gerenciadores de Certificados Digitais. Porém, exitem implementações que são acopladas a uma marca e modelo especifico de MSC. Esse acoplamento faz com que a Autoridade Certificadora que usa esse SGC se torne refém das empresas que produzem o MSC, característica pouco atraente para SGCs Governamentais. Os SGCs Ywapa e Ywyra foram desenvolvidos para serem independentes de marca e modelo de MSC, através da *API Openssl Engine* (OpenSSL, 2013). Dessa forma, eles podem se comunicar com qualquer MSC que disponibilize uma implementação dessa API.

O controle de acesso do Ywapa e do Ywyra é desenvolvido para ser resistente à corrupção de seus usuários e as potencias perdas de credenciais de acesso. Isso é feito através de uma autenticação multi-fator $m$ de $n$. Esse tipo de autenticação utiliza um algoritmo de compartilhamento de segredo, para dividir um segredo em $N$ partes (SHAMIR, 1979). Sendo que, dessas $N$ partes, são necessárias $M$ partes para reconstituir o segredo completo, tal que

$M \leq N$.

Por exemplo, considere um segredo compartilhado, tal que $M = 2$ e $N = 3$, distribuído entre $N$ usuários. Se um usuário se corromper, ele não poderá recuperar o segredo, pois serão necessárias duas partes do segredo. Se uma parte do segredo for perdida, ainda será possível recuperar o segredo, pois restariam duas partes do segredo. Além da autenticação $M$ de $N$, os SGCs Ywapa e Ywyra também exigem o uso de certificados digitais e *smartcards* para a autenticação individual dos seus usuários.

Por fim, o sistema de Backup se destaca pois é implementado pelo próprio SGC. As informações exportadas da base de dados do Ywapa e do Ywyra são cifradas e vinculadas aos perfis de usuários que as exportaram, através de um certificado digital e uma assinatura digital, respectivamente. Dessa forma, é possível garantir a origem do backup, através da assinatura digital, e que ele só poderá ser aberto pelo custodiante do certificado usado para cifrá-lo.

### 3.4.2 Sistema Gerenciador de Certificados Digitais da ICPEdu

O Sistema Gerenciador de Certificados Digitais da ICPEdu (SGCI) foi desenvolvido pela Rede Nacional de Ensino e Pesquisa (RNP), como parte da sua Infraestrutura de Chaves Públicas para Ensino e Pesquisa (ICPEdu), que consiste em:

> *"A Infraestrutura de Chaves Públicas para Ensino e Pesquisa (ICPEdu) consiste na implantação de uma infraestrutura de criação de certificados digitais e chaves de segurança, aplicados em autenticação, assinatura digital e sigilo, dentro do ambiente das Instituições Federais de Ensino Superior (Ifes), Unidades de Pesquisa (UPs) e demais instituições de ensino"* (RNP, 2013).

O SGCI é um sistema gerenciador de certificados digitais de código aberto, implementado em PHP como uma aplicação *Web*, para ser executada através de um navegador de internet. Também desenvolvido pelo Laboratório de Segurança em Computação (LabSEC), o sistema já se encontra na sua segunda versão, lançada em Março de 2013 (LabSEC, 2013). Suas principais característica são a simplicidade e robustez. Durante seu desenvolvimento, diversos trabalhos acadêmicos foram desenvolvidos, propondo melhorias de segurança, usabilidade e suporte parcial ao CMP (KOHLER; JUNIOR, 2007; COSTA, 2012; SILVéRIO, 2011).

Com uma interface bem intuitiva, o SGCI permite criar e gerenciar Autoridades Certificadoras (Raízes e Subordinadas) e Autoridades de Regis-

tro. Sendo que, na sua última versão, é possível estabelecer vínculos de confiança entre ACs e ARs de dentro e fora do sistema. Seu sistema de controle de acesso é baseado em login e senha, sendo que um perfil, chamado Criador, pode cadastrar novos usuários e autorizar seu acesso às funções de uma Autoridade. Além disso, ele tem suporte a MSCs de diversas marcas e modelos, Sistemas de Logs e Backup com proteção de leitura.

### 3.4.3 Outros Sistemas Gerenciadores de Certificados Digitais

Além das aplicações de SGC com as quais o LabSEC se envolveu, foram analisados os SGCs *EJBCA*, *Xca*, *Tinyca* e *Gnomint*. A análise desses sistemas foi feita de forma mais simples, através da avaliação das funcionalidades providas por suas interfaces gráficas e previstas pelos seus manuais.

O EJBCA é um sistema gerenciador de certificados digitais desenvolvido pela *PrimeKey*. Ele é implementado em Java Web, baseado nos paradigmas de programação de orientação a objetos e componentes. Ele é uma ferramenta robusta, feita para alta performance, que pode ser configurada para atender as necessidades de vários tipos de Autoridades Certificadoras, podendo ser usado sozinha (*stand-alone*) ou integrada a outros sistemas (PrimeKey, 2013).

Dentre as aplicações de mercado avaliadas, o EJBCA é a mais complexa. Ele possui diversas opções de configuração e é flexível em todos os seu módulos. Em sua instalação padrão, foi observado que suas funções seguem as especificações feitas pelo CMP, porém a aplicação também é compatível com o protocolo de gerenciamento de certificados digitais SCEP (Simple Certificate Enrollment Protocol). Ao criar a AC, na instalação padrão, uma AR também é criada. Em tempo de execução, é possível criar novas ACs e executar as funções de emissão e revogação de certificados digitais, sendo que, para isso, o uso da AR é obrigatório.

As funções da AR incluem as funções de inicialização, certificação e revogação, previstas pelo CMP. Para criar vínculos com ARs externas é necessário que uma API, de acesso ao banco de dados da AR, seja implementada e importada no sistema de AC. Então, o sistema de AC pode consultar o banco de dados da AR periodicamente, em busca de novas mensagens CMP/SCEP. O controle de acesso utiliza certificados digitais, sendo possível cadastrar novos usuários e configurar a autorização de acesso de forma modular. O EJBCA também inclui sistema de Logs e Backup, sendo o segundo feito através de linha de comando.

No repositório de aplicações do Ubuntu é possível encontrar diversos sistemas gerenciadores de certificados digitais gratuitos. Dentre eles, foram

avaliados os SGCs *Xca*, *Tinyca* e *gnoMint*. Esses sistemas são implementações simples, com interface gráfica, que permitem gerenciar Autoridades Certificadoras. Todos eles implementam as funções de criação de requisição de certificado, emissão de certificados digitais, revogação de certificados digitais e emissão de LCRs. Todas as aplicações testadas são offlines e não implementam nenhum protocolo de gerenciamento de certificados digitais ou suportam um módulo de segurança criptográfico.

O controle de acesso dos SGCs *Xca* e *Gnomint* são baseados em login e senha para base de dados. Sendo que, para cada base de dados, é possível gerenciar várias Autoridades Certificadoras. O SGC *Tinyca* também implementa um controle de acesso baseado em login e senha, porém, cada AC possui sua própria credencial. Como suas bases de dados são implementas sobre o sistema de arquivos do sistema operacional, o backup consiste na replicação dos arquivos da base de dados.

## 3.5 DISCUSSÃO

Devido a complexidade dos requisitos funcionais e não funcionais previstos nas recomendações da IETF, procurou-se analisar os SGCs com base em suas principais funções. O conjunto dessas funções compõem o SGCs como um todo. A escolha dessas funções foi feita para salientar os principais desafios da emissão de grandes quantidades de certificados digitais.

### 3.5.1 A Autoridade de Registro

De forma geral, a função da AR é interfacear a comunicação entre Autoridades Certificadoras e seus Usuários Finais. Esse interfaceamento inclui diversas funções essenciais para a segurança do SGC, como: a identificação e autenticação física do requerente do certificado digital; a validação e edição de requisições de certificado digital; a aprovação ou rejeição das requisições; a inicialização de dispositivos criptográficos; e o envio de notificações para o Titular do Certificado (e.g., email, telefone, SMS, etc.).

Apesar da Autoridade de Registro ser especificada como uma entidade independente da entidade de AC, foi observado que nenhuma das aplicações analisadas possuem uma implementação independente para o sistema de Autoridade de Registro. O EJBCA é a aplicação mais próxima de uma implementação independente do sistema de AR, pois fornece a AR em um componente bem definido, coeso e desacoplado do resto do sistema. Porém, na instalação padrão do EJBCA, a AR é instalada junto com o sistema de AC,

criando um acoplamento entre essas entidades em tempo de execução.

O SGCI apresenta uma separação clara, em tempo de execução, entre as funções da AC e AR. Porém, ele não provê nenhum meio de instalação separada para os sistemas de AC e AR. Atualmente, ele possui suporte ao CMP, portanto ele é capaz de interagir com ACs e ARs externas. Dessa forma, é possível separar os sistemas de AC e AR criando duas instâncias do SGCI, uma para cada tipo de autoridade. Os demais SGCs avaliados possuem as implementações das funções de AC e AR fortemente acopladas, ou então, simplesmente não implementam o sistema de AR, embutindo suas funções no sistema de AC e reduzindo, assim, sua coesão.

A separação bem definida, coesa e desacoplada, das funções de AC e AR é benéfica para a ICP, pois reduz a superfície de contato das ACs com o meio externo e melhora a rastreabilidade das ações tomadas durante a execução das suas funções. Além disso, com ACs e ARs independentes, é possível escalar os serviços da ICP através da implantação de novas ARs e não novas ACs. Isso é importante pois, quando novas ACs são criadas, a organização estrutural e lógica da ICP é modificada, enquanto a criação de novas ARs possui um impacto menor nessa organização.

Esse problema com a separação das funções de AC e AR pode ser atribuído à pouca ênfase dada à AR pelas especificação da ICP X.509. No modelo proposto pelo CMP, a AR se encontra no mesmo ambiente da AC e não possui funções definidas especificamente para ela. Ela é tratada como um Usuário Final perante a AC e como uma AC perante o Usuário Final. Ou seja, a AR é apresentada apenas como um gargalo de requisições que redireciona as requisições dos Usuários Finais para a Autoridade Certificadora. Essa abstração pode até simplificar o protocolo de gerenciamento de certificados digitais. Porém, quando usada como modelo de desenvolvimento para Sistemas Gerenciadores de Certificados Digitais, pode causar problemas de interpretação que reduzem a coesão e aumentam o acoplamento das funções de AC e AR, como foi observado nos SGCs analisados.

### 3.5.2 O Agente de Registro

A maioria das funções da AR ocorre de forma automática. Porém, algumas delas precisam ser executadas por um usuários do Sistema de AR. Por exemplo, a função de aceitação ou rejeição de requisições de certificados digitais está presente no SGCI, no SGC de mercado e no *EJBCA*. Independente da implementação, sempre há um usuário responsável por essa função. Esse usuário pode ser chamado de funcionário da AR, operador da AR ou então, da forma como é chamado na ICP-Brasil, Agentes de Registro (AGR).

### 3.5.3 Requerente do Certificado

Outra característica notada nos SGCs avaliados, foi que, apesar do requerente ser especificado como uma entidade independente, a AR é quem costuma fornecer a aplicação pra ele executar suas funções. Como, por exemplo, a geração de chaves criptográficas, a geração de requisições de certificado e a instalação do certificado digital em sua máquina ou *token* criptográfico.

### 3.5.4 Os Recursos do SGC

As ACs e ARs precisam de recursos para realizar o armazenamento de dados e para utilizar as funções criptográficas com sua chave privada. Esses recursos estão presentes em todas as aplicações analisadas, inclusive nas mais simples. O armazenamento de dados é feito em bancos de dados relacional, enquanto as funções criptográficas são fornecidas por um provedor criptográfico. A única diferença, entre as aplicações, é que as aplicações mais simples utilizam provedores criptográficos que armazenam suas chaves privadas cifradas em disco, enquanto as aplicações mais complexas - EJBCA, SGCI, Ywapa e Ywyra - utilizam provedores criptográficos embarcados em hardware especializado, como os Módulos de Segurança Criptográficos.

O armazenamento de dados é essencial para o funcionamento correto das Autoridades Certificadoras e de Registro. Por exemplo, certificados no formato *X.509* possuem um campo de número serial, que deve ser preenchido com um valor único no contexto da AC emissora (COOPER et al., 2008). Para isso, a AC costuma usar um contador que é atualizado a cada emissão de certificado digital. A AC não pode perder esse contador, pois poderia emitir certificados com o mesmo número serial, o que causaria problemas em caso de revogação. Da mesma forma, Autoridades de Registro precisam manter armazenados diversos dados para executar corretamente suas funções.

Autoridades Certificadoras e de Registro estão diretamente envolvidas com procedimentos criptográficos. Principalmente, com a realização e validação de assinaturas digitais. Normalmente, esses procedimentos devem ser feitos através de algoritmos normalizados com implementação homologadas. Para isso, as ACs e ARs costumam utilizar um conjunto de hardware e software específico, conhecidos por Módulo de Segurança Criptográfico (MSC), ou em inglês, *Hardware Security Module* (HSM).

O MSC é um provedor criptográfico, que provê funções de cifração e decifração de dados. Seu diferencial se encontra na definição de um perímetro criptográfico lógico-físico, do qual o MSC garante que as chaves criptográficas gerenciadas por ele não sairão em claro (MARTINA; SOUZA; CUSTO-

DIO, 2007). Ou seja, o MSC é capaz de gerar chaves criptográficas e fornecer funções criptográficas, que utilizam essas chaves, para o meio externo, sem dar acesso direto às chaves.

A interação entre o módulo de segurança criptográfico e o mundo externo é feita através de chamadas de funções para *APIs* padronizadas - como o *PKCS#11* (RSA Laboratories, 2009), *OpensSSL Engine* (OpenSSL, 2013) e *Java Cryptography Architecture* (JCA) (SUN, 2002) - que são implementadas e distribuídas pelos fabricantes dos MSCs. Além dessas interfaces, os MSCs podem oferecer uma aplicação própria, para gerenciamento de suas chaves criptográficas. Essa aplicações normalmente são usadas para gerar as chaves criptográficas e liberá-las para uso. No sistema de AC e de AR, as funções do provedor criptográficos são usados, principalmente, pelas funções do Módulo Servidor. Mas, algumas funções do Módulo Gerenciador, como a função de Criação de Autoridade, precisam ter acesso ao provedor criptográfico.

### 3.5.5 O Gerenciamento de Autoridade

Nas análises realizadas, notou-se a existência de diversas funções similares, implementadas em diversos dos SGCs observados, que não são especificadas pelo PKIX da IETF. Essas funções estão relacionadas com a gerência das entidades ACs e ARs, diferente das funções previstas pelos protocolos CMC e CMP, que especificam as funções de operação dessas entidades. Essas funções foram reunidas e listadas na Tabela 1, e relacionadas com as aplicações que as implementam. As funcionalidades marcadas com "x" indicam que ela foi implementada pela aplicação, enquanto as marcadas com "-" indicam que a funcionalidade foi parcialmente implementada ou que ela pode ser atingida através de aplicações externas, como *scripts* fornecidos junto com a aplicação principal.

Todas as aplicações analisadas implementam a função de criação de autoridade certificadora. Seu processo é semelhante em todas as aplicações, cria-se uma chave, um certificado e as configurações relativas à AC. Dentre essas aplicações, foram verificadas dois momentos distintos da criação da AC. Três dos SGCs analisados precisam criar uma AC antes de liberar qualquer função do SGC, enquanto os demais SGCs permitem a execução de funções, não relacionadas à AC, antes da criação da AC, como, por exemplo, o cadastro de usuários, a visualização de logs, o cadastramento de MSCs e criação de chaves.

A criação de AR já se mostra uma funcionalidade mais rara entre as aplicações analisadas. Apesar de todos os sistemas que implementam o sistema de AR terem que criar a AR em algum momento, apenas o SGCI permite

Tabela 1 – Relação Função x Aplicação.

| Função/Aplicação | EJBCA | SGCI | João de Barro |
|---|---|---|---|
| Criação de AC | x | x | x |
| Criação de AR | - | x | |
| Suporte Múltiplas ACs | x | x | x |
| Suporte Múltiplas ARs | - | x | |
| Controle de Acesso | x | x | x |
| Backup Completo | - | x | x |
| Backup de AC | - | | x |
| Backup de AR | | | |
| Importação de AC | - | | x |
| Importação de AR | | | |
| Registro de Logs | x | x | x |
| Perfil de Auditoria | x | | |
| Vínculo AC-AR | x | x | |
| Cadastro AGR | x | x | |
| Suporte MSC | x | x | x |

a sua criação de forma autônoma da AC. Ou seja, os outros SGC criam uma AR no momento da criação de AC ou na instalação da aplicação, que é o caso do EJBCA, em sua instalação padrão.

Todos os SGCs analisados suportam a criação e gerenciamento de múltiplas ACs em uma única instância da aplicação, enquanto apenas o SGCI suporta a criação e gerenciamento de múltiplas ARs por instância de aplicação. O EJBCA até permite a criação de múltiplas ARs, mas apenas através de ferramentas externas à sua instalação padrão.

Todos os SGCs avaliados implementam algum tipo de controle de acesso, seja por login e senha ou segredo compartilhado. Além disso, a maioria deles implementam perfis de usuários que restringem o acesso à determinadas funções do SGC, como, por exemplo, o acesso às funções de autoridades certificadoras diferentes.

Todos SGCs também implementam algum tipo de criação e recuperação de backup do sistema. Porém, apenas o SGCI, Ywapa e Ywyra possuem essas funcionalidades implementadas na própria aplicação. O EJBCA utiliza mecanismos externos para gerar e recuperar backups do sistema.

As funcionalidades de backup e recuperação de backup de AC aparece nos SGCs EJBCA, Ywapa e Ywyra. Porém, o EJBCA provê essa função através de *script shell* externo à aplicação do SGC. As funcionalidades de backup e recuperação de backup de AR não foram verificadas em nenhum

dos SGCs avaliados.

## 3.6 CONCLUSÃO

Neste capítulo, foram apresentados as principais especificações utilizadas no gerenciamento de certificados digitais. Essas especificações são as RFCs da ICP X.509, que tratam das estruturas de dados, algoritmos e protocolos utilizados em todo ciclo de vida dos certificados digitais. Também foram apresentadas as aplicações que implementam essas especificações com o objetivo de prover uma plataforma de gerenciamento de certificados digitais para Autoridades Certificadoras e Registradoras.

Vale ressaltar que o acesso à esse tipo de aplicação é bem limitado, devido à sua natureza sigilosa. Portanto, a análise se manteve em aplicações que o autor teve acesso, seja através dos convênios do Laboratório de Segurança em Computação com instituições externas ou de implementações de código aberto. Das aplicações avaliadas, destacam-se três - Ywapa, Ywyra e SGCI - que são soluções utilizadas em ambientes de produção de alta segurança.

O objetivo da avaliação feita neste capítulo é destacar as etapas do ciclo de vida do certificado digital que aparecem nas implementações de SGC, mas não nas especificações do modelo X.509. Também são destacados os problemas trazidos pela abstração proposta pelo modelo X.509 para as entidades que participam do ciclo de vida do certificado digital, principalmente, a abstração para Autoridades Registradoras. Com essa análise, cria-se uma base para a especificação do modelo proposto no próximo capítulo, que visa complementar o modelo de gerenciamento de certificados digitais proposto pelas RFCs da ICP X.509.

# 4 NOVO MODELO DE SGC

## 4.1 INTRODUÇÃO

Neste capítulo é apresentado um novo modelo de gerenciamento de certificados digitais. Esse modelo é uma extensão do modelo proposto pelo PKIX da IETF. Ele apresenta uma visão mais detalhada das entidades que participam do ciclo de vida do certificado digital, expandindo suas funções e revelando os recursos utilizados, por elas, para gerenciar certificados digitais. O detalhamento é necessário para que se possa propor a distribuição do processo de emissão em larga escala de certificados digitais, mantendo o controle centralizado da chave privada da autoridade certificadora.

A construção desse modelo foi baseado nas análises feitas no capítulo anterior, sobre as especificações das RFCs e as aplicações de gerenciamento de certificados digitais.

## 4.2 SEPARAÇÃO AC E AR

Uma das propostas deste trabalho é definir a AR como uma entidade obrigatória no Sistema Gerenciador de Certificados Digitais. A separação bem definida da AC e da AR é benéfica para o SGC, pois reduz a superfície de contato da AC com o meio externo e melhora a rastreabilidade das ações tomadas durante a execução das suas funções (esse assunto é aprofundado no Capítulo 6). Para representar essa alteração, é proposto um diagrama relacional, ilustrado pela Figura 17, que é baseado no modelo da ICP *X.509*.

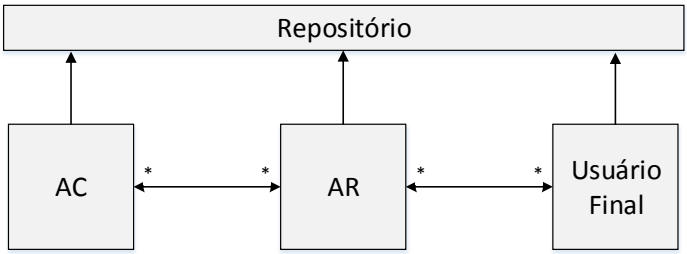


Figura 17 – Diagrama Relacional das Entidades da ICP *X.509*

A especificação do modelo em diagramas relacionais dá uma visão mais clara de como são organizadas as entidades da ICP. Por exemplo, a relação entre AC e AR é de muitos para muitos, informação que não é clara no diagrama ilustrado pela Figura 16. Além disso, essa forma de representação mantém a semântica dos canais de comunicação expressos no modelo da ICP *X.509*.

Atualmente, não existe nenhuma especificação formal, nas RFCs da ICP *X.509*, para o Agente de Registro e suas funções. Também não há nenhuma especificação para certas funções do requerente de certificado, como: a geração de chaves criptográficas, a geração de requisições de certificado e a instalação do certificado digital em sua máquina ou *token* criptográfico. Para melhor representar esses usuários e funções, foram adicionadas duas novas entidades no diagrama relacional proposto anteriormente, como mostrado na Figura 18.

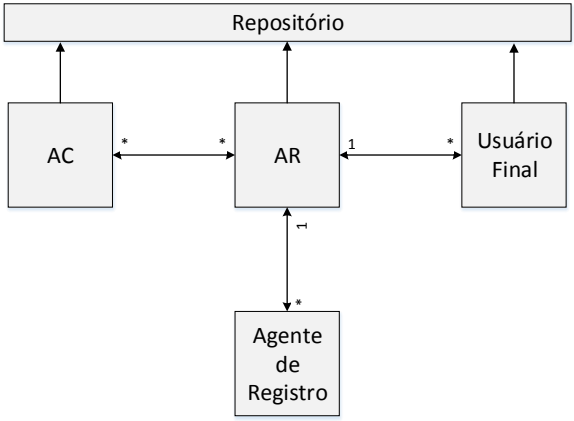


Figura 18 – Inclusão do AGR no Diagrama Relacional.

Essas entidades são constituídas de aplicações chamadas, neste trabalho, de Módulo Público e Módulo AGR. A especificação desses módulos é importante, pois, durante o desenvolvimento de um sistema de AR, os desenvolvedores devem estar cientes da existência deles e das suas funções. A especificação desses módulos é feita na última Seção deste Capítulo, sendo eles ilustrados no diagrama da Figura 21.

## 4.3 AUTORIDADE CERTIFICADORA

Conforme apresentado no capítulo anterior, diversas funções similares, não especificadas pelas RFCs, são implementadas por diversos SGCs. Essas funções mostram-se envolvidas com o gerenciamento da AC, enquanto as funções especificas pelas RFCs se mostram envolvidas com a operação da AC.

Para representar a separação dessas funções no modelo de SGC, foram especificados dois módulos: o Módulo Gerenciador e o Módulo Servidor. O Módulo Servidor representa o conjunto de funções relacionadas com a operação da AC, ou seja, as funções previstas pelo CMC ou CMP. O Módulo Gerenciador, por sua vez, representa o conjunto de funções relacionadas ao gerenciamento de ACs, e inclui todas as funções listadas na Tabela 1.

Considerando a motivação de manter os sistemas de AC e de AR separados e bem definidos, propõe-se que cada um deles deva possuir seus próprios módulos de gerenciamento e de serviço, conforme ilustra a Figura 19. Ou seja, o sistema de AR possui um Módulo Gerenciador de AR (GAR) e um Módulo Servidor de AR (SAR), enquanto o sistema de AC possui um Módulo Gerenciador de AC (GAC) e um Módulo Servidor de AC (SAC), sendo que, para executar as funções de gerenciamento de certificado digital, apenas os módulos servidores, de AC e de AR, se comunicam.

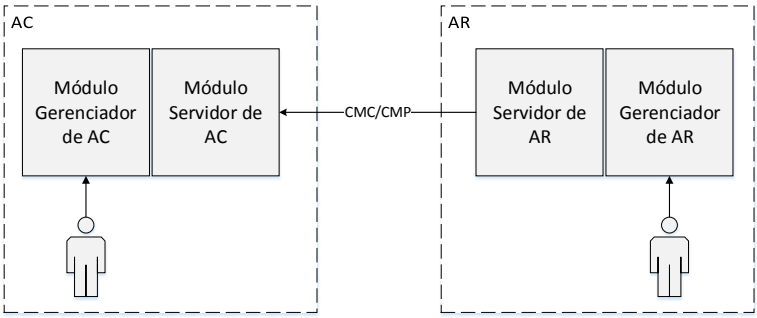


Figura 19 – Módulos Gerenciadores e Servidores.

O ambiente de AC é formado por, pelo menos, um Módulo Gerenciador de AC, um Módulo Servidor de AC, um MSC e um Banco de Dados. O banco de dados é responsável pelo Espaço de Armazenamento de AC, enquanto o MSC é o provedor criptográfico da AC. Sendo que os Módulos Gerenciadores e Servidores compartilham os recursos da AC que operam. Além

dos módulos, são ilustrados, na Figura 20, o usuário que opera o Módulo Gerenciador. Esse usuário representa, pelo menos, uma pessoa que opera o módulo através de uma interface para humanos, normalmente uma interface gráfica. O Módulo Servidor de AC não possui nenhuma interação humana, portanto, opera de forma automática através de um protocolo de gerenciamento de certificados digitais.

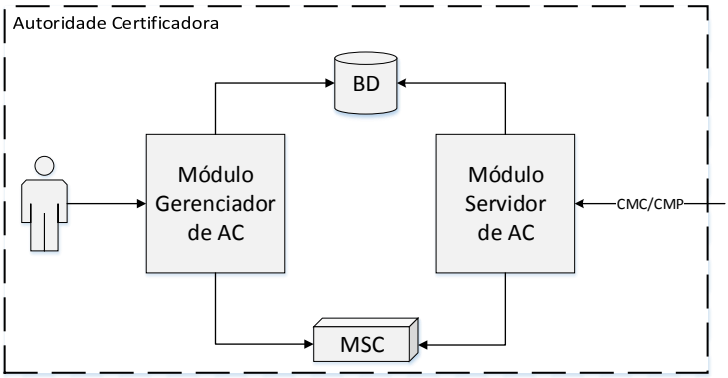


Figura 20 – Ambiente da AC.

Todas as funções especificadas para o modelo estendido de SGC estão listadas a seguir, relacionadas com os módulos onde devem ser implementadas. A especificação de como cada uma dessas funções devem ser implementadas são especificadas na próxima Seção.

**Definição 6 (Módulo Gerenciador de AC)** O Módulo Gerenciador de AC é constituído pelo conjunto de pessoas, máquinas e aplicações que interagem para fornecer as funções de gerenciamento de Autoridade Certificadora, que incluem:

- **Criação de AC:** A criação de AC consiste na geração das suas chaves criptográficas, seu certificado digital, seu espaço de armazenamento e na emissão de uma LCR inicial, vazia;

- **Gerenciamento do Vinculo com AR:** O gerenciamento do vínculo com AR inclui o cadastramento de informações que permitem autenticar as solicitações provenientes de um AR confiável; e a ativação e desativação de vínculos já cadastrados;

- **Configuração do Intervalo de Emissão de LCR:** A configuração do intervalo de emissão de LCR consiste na definição de um intervalo de tempo que será usado para definir a frequência de emissão de LCRs. Esse intervalo deve ser definido logo após a criação da AC, para ser usado na primeira LCR emitida;

- **Gerenciamento de Backup:** O gerenciamento de backup completo consiste na geração e recuperação de backups de todo Sistema de AC; O gerenciamento de backup de AC consiste na geração e recuperação de backups de uma única AC;

- **Controle de Acesso ao Módulo Gerenciador:** O controle de acesso ao módulo gerenciador inclui as funções de cadastramento, autorização e autenticação dos usuários que podem acessar o sistema. Essas funções devem ser executadas antes da criação de AC;

- **Configuração do Acesso aos Recursos da AC:** A configuração do acesso aos recursos da AC consiste no cadastramento das informações necessárias para que o Sistema de AC possa acessar o espaço de armazenamento da AC e o seu provedor criptográfico. Essas informações devem ser compartilhadas com o Módulo Servidor de AC.

**Definição 7 (Módulo Servidor de AC)** O Módulo Servidor de AC compreende o conjunto de pessoas, máquinas, e aplicações que interagem para fornecer as funções estabelecidas pelo protocolo CMP para a entidade AC. Também é incluso, nesse conjunto de funções, a função de emissão automática de LCRS, que deve obedecer a frequência de emissão definida pelo Módulo Gerenciador da AC.

## 4.4 A AUTORIDADE DE REGISTRO

Da mesma forma que a AC, a autoridade de registro possui um conjunto de funções relacionadas ao seu gerenciamento que não são especificadas pelas RFCs da ICP X.509. Esse conjunto de funções é similar ao conjunto de funções de gerenciamento de AC. A representação da separação dessas funções é feita da mesma forma que na AC, através de Módulos Gerenciadores e Módulos Servidores. Os Módulos Servidores de AR acumulam as funções previstas pelo CMP, enquanto os Módulos Gerenciadores de AR acumulam as funções de gerenciamento de AR.

Da mesma forma, o ambiente de AR é formado por, pelo menos, um Módulo Gerenciador de AR, um Módulo Servidor de AR, um MSC e um

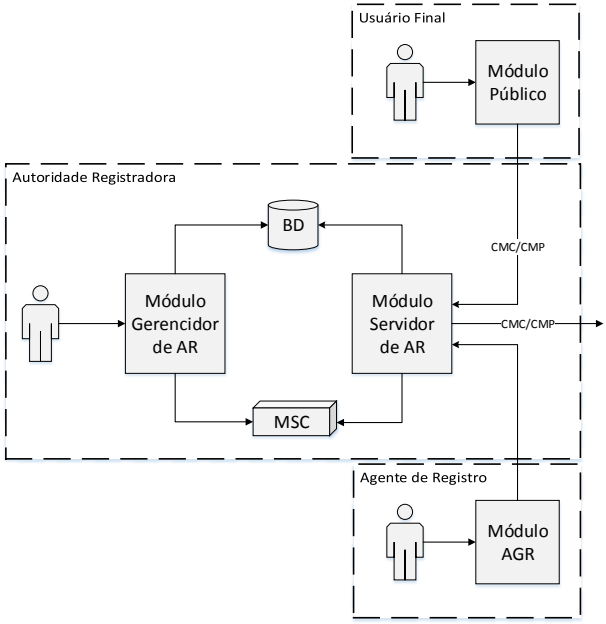


Figura 21 – Ambiente da AR.

Banco de Dados. O banco de dados é responsável pelo Espaço de Armazenamento da AR, enquanto o MSC é o provedor criptográfico da AR. Porém, a Figura 21 também inclui os Ambientes de Usuário Final e de AGR, com o Módulo Público e o Módulo AGR, respectivamente. Esses ambientes se encontram próximos ao ambiente de AR, pois são os Sistemas de AR que costumam fornecer as aplicações para o Usuário Final e para o Agente de Registro executarem suas funções.

Além dos módulos, são ilustrados, na Figura 20, os usuários que operam o Módulo Gerenciador, o Módulo Público e o Módulo AGR. Esses usuários representam, pelo menos, uma pessoa que opera o módulo através de uma interface para humanos, normalmente uma interface gráfica. O Módulo Servidor de AR não possui nenhuma interação humana, portanto, opera de forma automática através de um protocolo de gerenciamento de certificados digitais.

Os Módulos Gerenciadores e Servidores compartilham os recursos da AR que operam. O Módulo Público se relaciona com o Módulo Servidor de AR através de um protocolo de gerenciamento de certificado digitais. Porém,

nessa comunicação, também podem ser transportadas mensagens relacionadas às regras de negócio da AR, não previstas pelo protocolo. A comunicação do Agente de Registro com a AR não possui uma padronização, portanto, pode ser feita através de um protocolo definido pela AR.

**Definição 8 (Módulo Gerenciador de AR)** O Módulo Gerenciador de AR é constituído pelo conjunto de pessoas, máquinas e aplicações que interagem para fornecer as funções de gerenciamento de Autoridade de Registro. Essas funções incluem a Criação de AR, o Gerenciamento de Vínculo com AC, o Gerenciamento de Backup, o controle de Acesso ao Módulo Gerenciador e a Configuração de Acesso aos Recursos da AR. Essa funções se assemelham às funções definidas para o Módulo Gerenciador de AC.

Além dessas funções, o Módulo Gerenciador de AR é responsável pelo Gerenciamento de Agentes de Registro, que inclui as funções de cadastramento das informações que permitem autenticar as solicitações provenientes de um Agente de Registro e a habilitação e desabilitação desses AGRs enviarem solicitações à AR.

**Definição 9 (Módulo Servidor de AR)** O Módulo Servidor de AR compreende o conjunto de funções estabelecido pelo protocolo CMP para a entidade AR. Além disso, o Módulo Servidor de AR pode prover as aplicações que representam o Módulo Público e o Módulo AGR.

**Definição 10 (Módulo Público)** O Módulo Público é constituído pelo conjunto de máquinas e aplicações que permitem o Usuário Final executar todas as funções básicas previstas pelo CMP para entidade Usuário Final interagir com uma Autoridade de Registro. Além dessas funções, são incluídas: a geração de chaves criptográficas, a geração de requisições de certificado (PKCS#10 ou CRMF) e a Instalação do Certificado Digital no dispositivo do Usuário Final (e.g., sistema operacional, *browser*, *smartcards*, etc.).

**Definição 11 (Módulo de Agente de Registro)** O Módulo de Agente de Registro é constituído pelo conjunto de máquinas e aplicações que permitem um Agente de Registro a realizar as seguintes funções:

- **Cadastramento de Usuários Finais:** o cadastramento do Usuário Final consiste no armazenamento de informações que permitem identificar o Usuário unicamente. Essas informações podem ser utilizadas no preenchimento dos campos da requisições de certificado;

- **Visualização das Solicitações:** o Módulo AGR deve permitir que o Agente de Registro realize buscas e visualize solicitações de emissão e revogação de certificados digitais;

- **Edição das Requisições de Certificado Digital:** o Módulo AGR deve permitir que o Agente de Registro edite os campos que serão incluídos na requisição de certificado digital do Usuário Final;

- **Aprovação ou Rejeição das Solicitações:** os Agentes de Registro devem ser capazes de aprovar ou rejeitar solicitações de emissão ou revogação de certificados digitais;

- **Solicitação de Revogação de Certificado:** os Agentes de Registro também devem ser capazes de solicitar a revogação de certificados digitais de Usuários Finais.

## 4.5 OS RECURSOS DO SGC

Por fim, para representar os recursos utilizados pelas ACs e ARs (Armazenamento de Dados e Provedor Criptográfico), foram inseridos no diagrama um Módulo de Segurança Criptográfico e um Banco de Dados. O primeiro já possui diversas padronizações relacionadas a sua construção e funcionamento. Porém, não há nenhuma especificação formal para o armazenamento de dados para ACs e ARs. Dessa forma, uma formalização desses dados é apresentada a seguir.

**Definição 12 (Espaço de AC)** As informações que uma AC deve armazenar podem ser expressas na forma da tupla $E_{ac} = (c, f_d, sn_{cert}, sn_{lcr}, E_{cert}, E_{lcr}, R, V_{ar}, i_{lcr})$, sendo $c$ o certificado digital da AC; $f_d$ a sua chave privada, ou as informações de acesso à chave privada; $sn_{cert}$ e $sn_{lcr}$ os próximos números seriais de certificado e LCR, respectivamente, tal que $sn_{cert}, sn_{lcr} \in \mathbb{N}^*$; $E_{cert}$ o conjunto de *certificados* emitidos pela AC; $E_{lcr}$ o conjunto de *LCRs* emitidas pela AC; $R$ o conjunto de *certificados revogados* pela AC; $V_{ar}$ o conjunto de Autoridades de Registro confiáveis vinculadas; e $i_{lcr}$ o intervalo de tempo entre as emissões de LCR.

Da mesma forma, Autoridades de Registro precisam manter armazenados diversos dados para executar corretamente suas funções. O conjunto desses dados é denominado de *Espaço de AR* ou *Espaço de Armazenamento da AR*, sendo sua definição dada a seguir.

**Definição 13 (Espaço de AR)** As informações que uma AR deve armazenar podem ser expressas na forma de uma tupla $E_{ar} = (c, f_d, U_{agr}, U_{req}, S_{cert}, S_{rev}, V_{ac})$, sendo $c$ o certificado digital da AR; $f_d$ a chave privada da AR, ou as informações de acesso à chave privada; $U_{agr}$ o conjunto de Agentes de Re-

gistro; $U_{req}$ o conjunto de Usuários Finais requerentes; $S_{cert}$ o conjunto de solicitações de emissão de certificado digital; $S_{rev}$ o conjunto de solicitação de revogação de certificado digital. $V_{ac}$ o conjunto de Autoridades Certificadoras vinculadas.

Essas definições suportam as funções básicas de AC e AR, o que não inclui as funções que envolvem custódia de chave privada. Além disso, os Sistemas de AC e AR também precisam armazenar outros dados, como credenciais do controle de acesso, Logs e configurações relacionadas às regras de negócio da AC ou AR. Esses dados serão especificados posteriormente, no Capítulo 5, pelo Modelo de Implementação.

## 4.6 CONCLUSÃO

Através do modelo proposto neste capítulo, fica evidente como o modelo proposto pela ICP X.509 abstrai diversas informações relevantes para o desenvolvimento de sistemas gerenciadores de certificados digitais. A especificação clara dos ambientes, módulos e recursos que forma um SGC permite que a implementação desses sistemas seja mais eficiente, garantindo melhor compatibilidade entre sistemas e melhor segurança. No próximos capítulo, esse modelo será usado como base para a arquitetura de um sistema gerenciador de certificados digitais online.

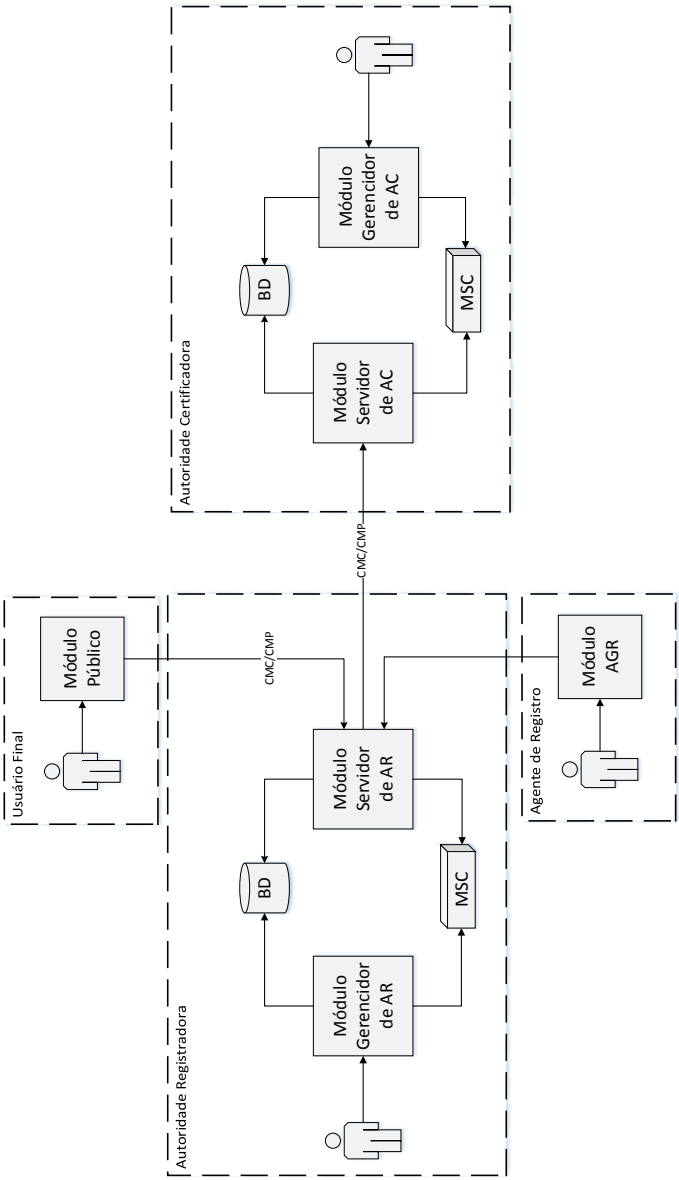


Figura 22 – Modelo completo de SGC.

# 5 SISTEMA ONLINE

## 5.1 INTRODUÇÃO

Neste Capítulo, é proposto um modelo de implementação para Sistemas Gerenciadores de Certificados Digitais Online, baseado na extensão do modelo da ICP *X.509*, proposto no Capítulo anterior. O modelo de implementação é preparado para ambientes distribuídos e de alta demanda. Além disso, ele propõe diversos mecanismos de segurança que cumprem com os requisitos esperados por ICPs Governamentais. Mas, de forma geral, ele pode ser usado em qualquer ICP que possua ambientes de alta demanda e que exijam níveis de segurança elevados.

## 5.2 VISÃO GERAL

A visão geral sobre o Modelo de Implementação é apresentada pela Figura 23, na forma de um diagrama de implantação que apresenta as cardinalidades das suas relações. Nesse diagrama, os módulos gerenciadores são representados por computadores de mesa, indicando que são operados por pessoas; enquanto os Módulos Servidores são representados por Servidores Web, que operam de forma automática e online. Esses Módulos compartilham os recursos das Autoridades pelas quais são responsáveis. Os perfis de usuários e controle de acesso do Módulo Gerenciador são propostos na Seção 5.3.

O Módulo Público e o Módulo AGR são Aplicações Web, fornecidas pelo Módulo Servidor de AR, acessadas através de computadores de mesa, pelos Usuários Finais e Agentes de Registro. Para proteger as funções do Módulo AGR, é definido um perímetro físico-lógico, chamado Instalação Técnica (IT), no qual os Agentes de Registro precisam estar inseridos para poder executar suas funções. Esse perímetro pode ser delimitado por diversos métodos que identificam o local físico, a máquina e a pessoa que está acessando o Módulo AGR. O Usuário Final, por sua vez, pode acessar o Módulo Público através de qualquer máquina que ele confie.

As relações entre os Módulos são as mesmas definidas no Capítulo 3, com exceção da relação entre o Módulo Gerenciador e o Módulo Servidor. Essa relação estabelece que cada Módulo Gerenciador pode se relacionar com mais de um Módulo Servidor e que cada Módulo Servidor pode se relacionar com mais de um Módulo Gerenciador. Ou seja, esses módulos passam a ser

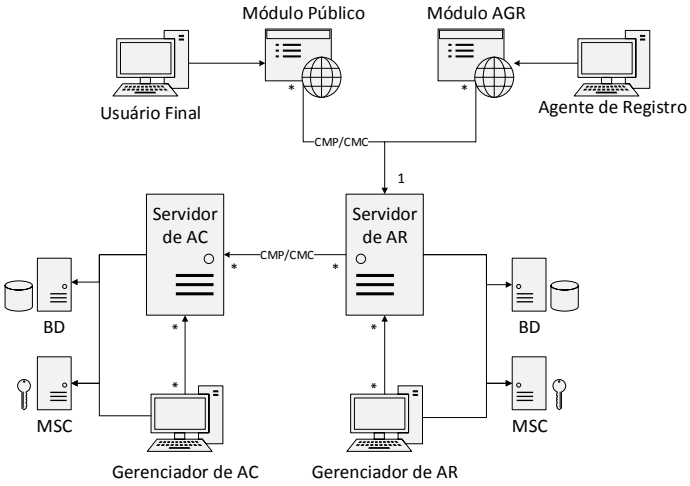


Figura 23 – Diagrama de Implantação do SGC.

independentes um do outro. Além disso, cada um desses módulos pode se relacionar de forma independente com as Autoridades de AC e de AR, como mostrado pelo diagrama relacional da Figura 24

Essas relações indicam que cada Módulo Gerenciador pode gerenciar mais de uma Autoridade, porém cada Autoridade só pode ser gerenciada por um Módulo Gerenciador; e que Cada Módulo Servidor pode operar mais de uma Autoridade e uma Autoridade pode ser operada por mais de um Módulo Servidor. Como os Módulos Gerenciadores e Servidores se relacionam de forma independente, então cada Autoridade do Módulo Gerenciador pode ser configurada em um ou mais Módulos Servidores, sendo que os Módulos Servidores podem operar Autoridades de Qualquer Módulo Gerenciador.

## 5.3 MÓDULOS GERENCIADORES

O Módulo Gerenciador de AC e o Módulo Gerenciador de AR compartilham várias das suas funções, portanto, a especificações delas foram reunidas nesta única seção. Para se referir a ambos os módulos, será usado o termo Módulo Gerenciador, e para se referir às ACs e ARs será usado o termo Autoridade. Quando for necessário, serão usados os nomes próprios desses Módulos e Autoridades.

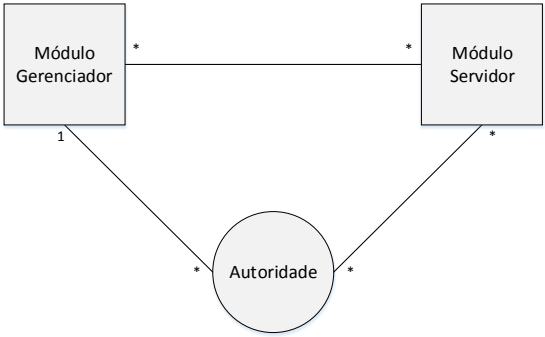


Figura 24 – Diagrama Relacional Módulos-Autoridade.

### 5.3.1 Controle de Acesso

O Módulo Gerenciador precisa implementar mecanismos de controle de acesso, para garantir que suas funções sejam acessadas apenas por pessoas autorizadas. Como o Módulo Gerenciador é capaz de gerenciar mais de uma Autoridade, também é necessário garantir que apenas os responsáveis por uma determinada Autoridade tenham acesso às funções dela. Além disso, para dificultar qualquer tentativa de corrupção dos usuários do módulo, é recomendado que nenhuma pessoa, sozinha, tenha acesso às suas funções.

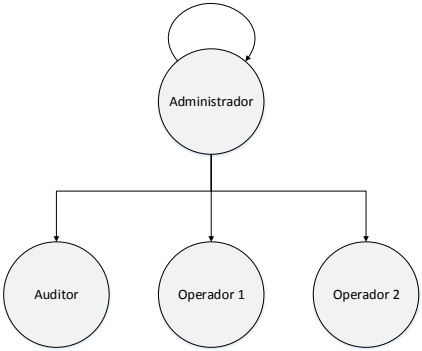


Figura 25 – Exemplo da ICP Interna do GAC e GAR

Para atender esses requisitos, são propostos três tipos de perfis de

usuário, são eles: Administrador, Operador e Auditor. Para cada tipo de perfil, é definido um conjunto de funções, pelas quais ele é responsável. Essa divisão é apresentada pela Tabela 2. A organização desses perfis é feita através de uma infraestrutura de chaves públicas interna ao Módulo Gerenciador, como ilustrado pela Figura 25. Cada certificado digital, nessa ICP, identifica um perfil de usuário. No topo da ICP, encontra-se o perfil de administração, com um certificado digital autoassinado. Logo abaixo, estão os perfis de auditoria e operação, com certificados emitidos pelo perfil de administração.

Tabela 2 – Relação Perfil de Usuário x Função do Módulo

| Função | Adm. | Ope. | Aud. |
|---|---|---|---|
| Criar Perfil | x | | |
| Gerar Backup Completo | x | | |
| Recuperar Backup Completo | x | | |
| Gerar Backup de Autoridade | x | x | |
| Recuperar Backup de Autoridade | | x | |
| Criar Autoridade | | x | |
| Gerenciar Servidores | | x | |
| Gerenciar Vínculo de Confiança | | x | |
| (GAC) Configurar Intervalo de LCR | | x | |
| (GAR) Gerenciar Instalações Técnicas | | x | |
| (GAR) Gerenciar Agentes de Registro | | x | |
| Geração de Relatórios Operacionais | | x | x |
| Visualização de *Logs* | | | x |
| Exportação de *Logs* | | | x |
| Visualização de *Logs* de Autoridade | | x | x |
| Exportação de *Logs* de Autoridade | | x | x |

Cada Módulo Gerenciador possui apenas um perfil de administração e um de audição. Enquanto o perfil de operação é associado a uma Autoridade, portanto, ele aparece em um número igual ao número de Autoridades gerenciadas pelo Módulo Gerenciador. As funções executadas pelos perfis de operação afetam apenas o contexto da Autoridade que eles gerenciam.

A autenticação de um perfil acontece através da prova de posse da chave privada do perfil. Para impedir que uma pessoa, sozinha, tenha acesso à essa chave, é proposto o uso de algoritmos de compartilhamento de segredo. Através desses algoritmos, é possível dividir uma chave simétrica $s$ em um conjunto de $n$ partes, e definir que são necessárias $m$ partes desse conjunto para recuperar a chave simétrica $s$, tal que $m \leq n$. Ou seja, se cada parte do segredo for entregue para uma pessoa, vão ser necessárias que $n$ pessoas

forneçam suas partes do segredo, para recuperar a chave simétrica *s*. Então, se essa chave simétrica for usada para cifrar a chave privada do perfil, só será possível autenticar o perfil se *n* pessoas apresentarem sua parte do segredo.

Para proteger cada parte do segredo, é sugerido que cada membro do perfil possua um certificado digital próprio, com uma chave privada armazenada em *smartcard*. Recomenda-se, ainda, que esse certificado digital tenha valor legal, para que o membro do perfil possa ser responsabilizado legalmente pelo uso da sua chave privada. Dessa forma, a parte do segredo pode ser armazenada pelo Módulo Gerenciador, desde que esteja cifrada com o par de chaves do membro responsável pela parte do segredo.

**Definição 14 (smartcard)** é um dispositivo semelhante ao módulo de segurança criptográfico. Ele permite que chaves criptográficas sejam geradas e usadas sem que saiam de um perímetro de segurança. A diferença se encontra em sua portabilidade. O *smartcard* é construído em um chip pequeno, normalmente colocado em um cartão de plástico para facilitar seu armazenamento em uma carteira. O uso do *smartcard* depende de uma leitora de cartões apropriada, que deve estar conectada à máquina onde o cartão será usado. Outro dispositivo utilizado para o mesmo fim é o *token* criptográfico, que pode ser utilizado diretamente em uma porta USB, sem a necessidade de uma leitora de cartões.

**Entradas:** Um perfil $p$, aberto; uma chave simétrica $f_s$;
**Saída:** O perfil $p$, fechado.

---

$K_{partes} \leftarrow split(f_s, |p.G|, |p.G|)$
para cada $g \in p.G \wedge k_{parte} \in K_{partes}$ faça
    $K'_{partes} \leftarrow split(k_{parte}, |g.N|, g.m)$
    para cada $n \in g.N \wedge k'_{parte} \in K'_{partes}$ faça
        $m.s := m.C.f_c(k'_{parte})$
    fim
fim
$P.f_d := f_s(P.f_d)$
$P.e := fechado$
return $P$

---

Algoritmo 5.1 – Algoritmo de Compartilhamento de Segredo

Os algoritmos de compartilhamento e restauração de segredo do perfil são exemplificados pelos Algoritmos 5.1 e 5.2. Nesses algoritmos, são usadas funções genéricas de compartilhamento e reconstrução de segredo. Também são usadas estruturas de dados para representar o perfil e os membros do perfil. Todas essas funções e estruturas são definidas a seguir.

**Entradas:** Um perfil $p$, fechado; e um conjunto $U$ de Usuários;
**Saída:** O perfil $p$, aberto.

```
para cada g ∈ p.G faça
        para cada m ∈ g.M faça
                f_d := U.find(m.C).f_d
                K'_partes ← f_d(m.s)
        fim
        K_partes ← join(K'_partes)
fim
f_s := join(K_partes)
p.f_d := f_s(p.f_d)
p.e := aberto
return p
```

Algoritmo 5.2 – Algoritmo de Restauração de Segredo do Perfil

**Definição 15 (Função split)** A função *split* recebe como entrada uma de tripla $(f_s, m, n)$, onde $f_s$ é um segredo (*chave simétrica*) e $m, n \in \mathbb{N}$, sendo $0 < m \leq n$. A função retorna um conjunto de tamanho $n$, contendo as partes do segredo, de forma que com $m$ partes é possível recuperá-lo (a chave simétrica).

**Definição 16 (Função join)** A função *join* recebe como entrada um conjunto de partes do segredo e retorna o segredo reconstituído $f_s$, se houverem partes suficientes conforme definido quando foi compartilhado.

**Definição 17 (Usuário)** Um usuário é definido como uma tupla $U = (C, f_d)$, sendo $C$ o certificado digital que o identifica e $f_d = f_c^{-1}$, tal que $f_c \in C$.

**Definição 18 (Membro de Grupo)** Um membro de grupo é definido por uma tupla $M = (C, s)$, sendo $C$ o certificado digital que identifica o membro; e $s$ a parte do segredo cifrada com uma chave pública $f_c$, tal que $f_c \in C$.

**Definição 19 (Grupo de Perfil)** Um grupo de perfil é uma tupla $G = (N, m)$, sendo $N$ um conjunto de membros de grupo; e $m$ o número necessários de membros para reconstruir um segredo compartilhado entre os membros do conjunto $N$, tal que $m \in \mathbb{N}^*$ e $m \leq |N|$.

**Definição 20 (Perfil)** Um perfil é uma tupla $P = (t, C, f_d, G, e)$, sendo que $t \in \{adm, ope, aud\}$, $C$ é um certificado digital da ICP interna do SGC; $f_d$ sua chave privada; $G$ o conjunto de Grupos do Perfil; e $e$ o estado do perfil, sendo $e \in \{aberto, fechado\}$. Quando o perfil está no estado *aberto*, a chave $f_d$ está decifrada, quando está no estado $fechado$ a chave está cifrada.

O algoritmo de compartilhamento de segredo recebe como entrada um Perfil $P$, aberto; e uma chave simétrica $f_s$, gerada aleatoriamente. O algoritmo se inicia com o compartilhamento da chave $f_s$ entre todos os grupos do perfil, de forma que sejam necessárias todas as partes do segredo para recuperar $f_s$. Então, para cada Grupo $g$ do perfil, compartilha-se sua parte do segredo entre todos os membros do grupo, de forma que sejam necessárias $m$ partes do segredo para recuperá-lo, tal que $m \in g$. Cada parte do segredo é, então, cifrada com a chave pública de um membro perfil $n$, tal que $n \in N$. Por fim, a chave privada do perfil deve ser cifrada com a chave simétrica $f_s$ e o estado do perfil alterado para $fechado$. Sendo a chave $f_s$ destruída no fim do processo.

O algoritmo de restauração de segredo do perfil recebe como entrada um Perfil $P$, fechado, e um conjunto $U$ de Usuários. Para cada grupo $g$, tal que $g \in G$ e $G \in P$, e para cada membro $n$, tal que $n \in N$ e $N \in g$, recupera-se o usuário $u$, tal que $u \in U$, identificado pelo mesmo certificado do membro $m$ (sendo $C_m = C_u$, tal que $C_m \in m$ e $C_u \in u$). Com a chave privada $f_d$ do usuário ($f_d \in u$), decifra-se a parte do segredo pertencente ao membro do grupo. Após decifrar pelo menos $n$ partes de segredos dos membros do grupo, tal que $n \in G$, recupera-se a parte do segredo do grupo. Após recuperar as partes de segredo de todos os grupos, recupera-se a chave simétrica $f_s$. Por fim, a chave privada $f_d$ do perfil e decifrada com a chave simétrica $f_s$ e seu estado é alterado para $aberto$.

É importante ressaltar que esses algoritmos não são de autenticação do perfil. Para realizar a autenticação, é necessário verificar se o certificado do perfil foi emitido pelo perfil de administração do Módulo Gerenciador. Para, então, verificar se a chave privada restaurada, pelo algoritmo de restauração de segredo, responde a um protocolo de desafio-resposta. Futuramente, na especificação das funções de backup, o certificado do perfil de administração precisará ser exportado do Módulo Gerenciador, para ser usado como âncora de confiança na restauração do backup.

### 5.3.2 Espaço de Armazenamento de Aplicação

No Capítulo 3 foram especificados os Espaços de Armazenamento de AC e AR. Porém, os dados previstos por suas definições não são suficientes para esse modelo de implementação. Outros dados precisam ser armazenados para que o SGC possa funcionar corretamente.

Por exemplo, nesse modelo de implementação, cada Autoridade possui um perfil de Operação responsável por ela. Além disso, cada Autoridade também possui um conjunto de Módulos Servidores que são autorizados a executar suas funções. Dessa forma, os dados necessários para realizar o

controle de acesso do perfil de operação, e os dados relacionados aos Módulos Servidores, também devem ser armazenados e relacionados com o Espaço da Autoridade. O conjunto desses dados é chamado de Espaço de Operação.

Além do Espaço de Operação, é preciso definir um espaço de armazenamento de Logs. Nesse espaço serão registrados todos os logs do sistema, dos quais apenas o perfil de Auditoria tem acesso. O conjunto desses dados é chamado de Espaço de Auditoria. Por fim, é necessário definir o Espaço de Administração, onde são armazenados todos os Espaços de Operação e o Espaço de Auditoria.

**Definição 21 (Espaço de Operação)** O Espaço de Operação é definido como uma tupla $e_{ope} = (p_{ope}, e_{aut}, S)$, sendo $p_{ope}$ um perfil de operação; $e_{aut}$ um espaço de Autoridade (AC ou AR); e $S$ um conjunto de Módulos Servidores.

**Definição 22 (Espaço de Auditoria)** O Espaço de Auditoria é definido como uma tupla $e_{aud} = (p_{aud}, e_{log})$, onde $p_{aud}$ é um perfil de auditoria e $e_{log}$ é um Espaço de Armazenamento de *Logs*.

**Definição 23 (Espaço de Administração)** O espaço de administração é definido como uma tupla $e_{adm} = (p_{adm}, E_{ope}, e_{aud})$, sendo $p_{adm}$ um perfil de administração; $E_{ope}$ um conjunto de Espaços de Operação; e $e_{aud}$ um Espaço de Auditoria.

Cada um dos espaços definidos deve ser protegido pelo perfil que o compõe, seja através de criptografia (cifração dos dados) ou apenas controle de acesso simples. Por exemplo, dado um Espaço de Operação $e$, só deve ser possível acessar $E_{aut}$ e $S$, tal que $E_{aut}, S \in e$, se o perfil $P$ for autenticado, tal que $P \in e$. O perfil $P$ não precisa ser protegido, pois seus dados sensíveis são cifrados com a chave pública do seu certificado.

O controle de acesso ao Espaço de Armazenamento de *Log* é diferente dos outros Espaços de Armazenamento. Como as funções de todos os perfis de usuário podem gerar *Logs*, então, a execução dessas funções devem ser capazes de escrever no Espaço de Armazenamento de *Logs*. Porém, a escrita deve ser controlada, para impedir a geração de *Logs* fraudulentos. Esse controle poderia ser feito através da autenticação dos perfis de usuário, porém, a principal função de um sistema de *Logs* é proteger a aplicação dos seus próprios usuários. Esse problema faz parte de um conjunto de problemas mais complexo, relacionados a sistemas de *Log* (SCHNEIER; KELSEY, 1998). Dessa forma, o registro de logs não se encontra no escopo deste trabalho.

### 5.3.3 Gerenciamento de Backups

O gerenciamento de backup consiste nas funções de geração e recuperação de backup. Sendo que existem três tipos de *backup*: os backups de administração, os backups de auditoria e os backups de operação. Cada um deles está relacionado com os espaços de armazenamento de mesmo nome. Para os três casos, os dados exportados devem ser protegidos contra leitura e modificação não autorizada. Então, sugere-se que cada *backup* seja cifrado com a chave pública do perfil que pode lê-lo, e assinado com a chave do perfil que o gerou. Levando isso em consideração, a definição dos dados que devem estar contidos no backup é dada a seguir.

**Definição 24 (Backup)** O backup pode ser representado por uma tupla $b = (p, D, Au, c)$, onde $p$ é um perfil; $D$ é um conjunto de dados cifrados com a chave pública do perfil $p$; $Au$ é uma assinatura digital feita sobre uma estrutura de dados contendo $p$ e $D$; e $c$ é o certificado digital necessário para verificar a assinatura $Au$.

Quando o backup for de um Backup de Operação, $p$ corresponde ao perfil de operação da Autoridade; e $D = (e_{aut}, S)$, sendo $e_{aut}$ o Espaço de Armazenamento da Autoridade e $S$ o conjunto de Módulos Servidores da Autoridade. Quando for um Backup de Auditoria, $p$ corresponde ao perfil de auditor; e $D = E_{log}$, sendo $E_{log}$ o Espaço de Armazenamento de *Logs*. Por fim, quando for um Backup for Completo, $p$ corresponde ao perfil de administração; e $D = (B_{ope}, b_{aud})$, sendo $B_{ope}$ um conjunto de Backups de Autoridade e $b_{aud}$ um backup de Auditoria. A seguir, são apresentados os algoritmos de geração e recuperação de backup, para cada um dos tipos propostos.

**Entradas:** um perfil $p$, aberto; um Espaço de Operação ou Auditora $x$
**Saída:** Um Backup de Ambiente $B$

---

$B.p := x.p$
$B.D := x.p.c.f_c(x - x.p)$
$B.Au := p.f_d(B)$
$B.c := p.c$
retorna $B$

---

Algoritmo 5.3 – Geração de Backup de Operação e Auditoria

A geração dos backups de operação e auditoria ocorrem de forma semelhante. Eles recebem como entrada um perfil $p$, aberto, representando o perfil que está gerando o backup; e um Espaço de Armazenamento $x$, do qual será feito backup. Sendo que os perfis que podem gerar um backup de ope-

ração ou auditoria são o de administração e o próprio perfil responsável pelo Espaço de Armazenamento.

A geração do backup de administração recebe como entrada um perfil $p$, aberto, representando o perfil que está gerando o backup; e um Espaço de Administração $e_{adm}$, do qual será feito backup. Sendo que apenas o perfil de administração pode gerar o backup de administração.

**Entradas:** um perfil $p$, aberto; um Espaço de Administração $e_{adm}$
**Saída:** Um Backup de Ambiente $B$

---

$B.p := e_{adm}.p$
para cada $e_{ope} \in e_{adm}.E_{ope}$
    $B_{ope} \leftarrow backup(p, e_{ope})$
fim
$D \leftarrow B_{ope}$
$D \leftarrow backup(p, e_{adm}.e_{aud})$
$B.D := e_{adm}.p.f_c(D)$
$B.Au := p.f_d(B)$
$B.c := p.c$
retorna $B$

---

Algoritmo 5.4 – Geração de Backup de Administração

A recuperação de backup só pode ser executada pelo perfil responsável pelo Espaço de Armazenamento do qual foi gerado o backup. O algoritmo recebe como entrada um backup $b$; um conjunto de usuários $U$, representando os usuários que vão autenticar o perfil responsável pelo backup; e um certificado $r$, representando o certificado de administração do Módulo Gerenciador de gerou o backup.

**Entradas:** Um Backup $b$, um conjunto de usuários $U$; e um certificado $r$
**Saída:** Um Espaço de Operação, Auditoria ou Administração $e$

---

se $(r.f_c(b.c.Au) = b.c - \{c.Au\})$ faça
    se $(b.c.f_c(b.Au) = B - \{b.Au, b.c\})$ faça
        $p := autentica(b.p, U)$
        $e.p := b.p$
        $e \leftarrow p.f_d(B.D)$
        retorna $e$
    fim
fim

---

Algoritmo 5.5 – Recuperação de Backup

O processo começa com a verificação da assinatura do certificado $c$ com a chave pública do certificado $r$, tal que $c \in b$. Essa verificação garante a origem do backup é de um Módulo Gerenciador confiável. Então, é verificada a assinatura $Au$ com a chave pública do certificado $c$, tal que $Au \in b$.

Essa verificação garante que o backup está íntegro e foi feito pelo perfil do certificado *c*. Após essas verificações, o perfil do backup deve ser autenticado para recuperar sua chave privada. Por fim, a chave privada do perfil é usada para decifrar o conteúdo cifrado do backup, e recuperar o Espaço de Armazenamento.

Se o backup restaurado for um Backup de Aplicação ou Auditoria, então o Módulo Gerenciador deve emitir um novo certificado para o perfil do espaço de armazenamento recuperado. É possível emitir esse certificado com a mesma chave pública do certificado anterior, assim não é necessário realizar um novo compartilhamento de segredo. Caso a restauração tenha sido de um Backup de Administração, o Módulo Gerenciador deve importar os certificados dos perfis como sua ICP interna. Os Espaços de Armazenamento de Operação e Auditoria permanecem no formato de backup, até que os operadores se autentiquem.

### 5.3.4 Criação de Autoridade

A criação de uma Autoridade, Certificadora ou de Registro, consiste na geração do seu par de chaves assimétricas, do seu certificado digital e do seu espaço de armazenamento. Esse processo é composto de cinco etapas: criação do par de chave, cadastro do par de chaves, criação da requisição de certificado, criação do certificado e importação do certificado. Essas etapas servem para Autoridades Certificadoras Raízes e Autoridades Intermediárias (AC ou AR), porém, possuem uma implementação diferente para cada uma delas. Essa diferença é resolvida com tratamentos diferenciados nas etapas de criação de requisição, criação do certificado e importação do certificado.

A seguir, são descritas as principais atividades realizadas em cada uma das etapas de criação de Autoridade. Essas descrições visam elucidar a função de criação de Autoridade Certificadora, apontando pontos que devem ser implementados ou não pelo Módulo Gerenciador.

**Etapa 1 (Criação do Par de Chaves):** Normalmente, a criação do par de chaves de uma Autoridade ocorre em um módulo de segurança criptográfico. Como os MSCs costumam ser disponibilizados junto com uma aplicação de gerência de chaves criptográficas, que possuem cerimônias específicas para esse conjunto de software e hardware, recomenda-se que o Módulo Gerenciador não implemente a função de criação de chaves criptográficas em MSC. Isso reduz a complexidade do Módulo Gerenciador, pois, para implementar essas funções, seria necessário que ele se adaptasse às normas relacionadas aos módulos de segurança criptográficos.

Portanto, mesmo sendo parte da criação de Autoridade Certificadora, a geração das chaves criptográficas deve ocorrer fora do Módulo Gerenciador. Mas, recomenda-se, que os responsáveis pela geração e custódia dessa chave sejam os operadores da Autoridade Certificadora que está sendo criada. Ao fim dessa etapa, algumas informações, como identificadores e senhas de acesso à chave, devem ser disponibilizadas para que sejam armazenadas pelo Módulo Gerenciador. Além disso, o MSC deve ser configurado para permitir que a chave seja usada pelo menos uma vez, de forma que o Módulo Gerenciador consiga realizar a assinatura necessária para a criação da requisição de certificado digital.

**Etapa 2 (Cadastro do Par de Chaves):** O cadastro do par de chaves consiste no armazenamento das informações de acesso à chave privada da autoridade, criada na Etapa 1. As informações que devem ser cadastradas diferem entre os tipos de APIs usadas pelos MSCs. Como é recomendado que o Sistema Gerenciador de Certificados Digitais seja independente de marca e modelo de MSC, é preciso que o Módulo Gerenciador seja capaz de armazenar os dados de acesso para, pelo menos, mais de um tipo de API.

Considerando os três tipos de APIs mais usadas, *PKCS#11*, *JCA* e *OpenSSL Engine*, as informações que devem ser armazenadas são:: o tipo de API utilizada; a implementação da API na forma de uma biblioteca dinâmica; e as informações de acesso à chave, conforme os requisitos do tipo de API escolhido. Essas informações devem ser salvas no Espaço de Armazenamento da AC, cifradas com a chave pública do perfil de operação responsável por ela. Ao fim do processo, o SGC pode realizar um teste de comunicação com o MSC, para garantir que as configurações estão corretas.

**Etapa 3 (Criação da Requisição de Certificado):** A criação da requisição de certificado da AC, consiste na exportação dos dados da Autoridade que serão usados na emissão do seu certificado. Atualmente, existem duas RFCs que padronizam as estruturas de dados que devem estar contidos na requisição de certificado, a RFC-4211 (CRMF) e a RFC-2986 (PKCS#10), sendo que a CRMF vem tomando o lugar do PKCS#10. Porém, recomenda-se que o Módulo Gerenciador seja capaz de criar requisições de certificados digitais nos dois formatos, para que se mantenha a compatibilidade com software legado. As informações que devem ser inseridas nessas requisições incluem (SCHAAD, 2005):

- o nome distinto da Autoridade;
- as informações sobre a chave pública da Autoridade;
- as extensões de certificado desejadas;

- e uma prova de posse da chave privada correspondente à chave pública da requisição.

Essas informações são equivalentes entre a CRMF e o PKCS#10, sendo suas representações feitas de forma diferente. A maior diferença se encontra na prova de posse da chave privada. Enquanto o PCKS#10 utiliza apenas uma assinatura digital, feita com a chave privada do requerente, a CRMF permite o uso de outras técnicas de prova de posse, úteis para atender às necessidade de usuários finais.

Porém, no contexto das Autoridades Certificadoras e de Registro, é recomendado o uso da assinatura digital, pela sua simplicidade e por não depender de uma interação prévia com a Autoridade Certificadora emissora. Portanto, após construir a requisição de certificado, deve-se realizar a assinatura sobre essas informações com a chave privada criada na Etapa 1. Sendo essa assinatura realizada conforme as regras do padrão escolhido.

Por fim, caso essa seja a criação de uma Autoridade Subordinada, a requisição deve ser armazenado pelo Módulo Gerenciador e exportada para o operador. Caso seja a criação de uma AC Raiz, a requisição deve ser mantida em memória, para que seja usada na próxima etapa.

**Etapa 4 (Criação do Certificado)** A criação do certificado digital da Autoridade pode ocorrer de duas formas diferentes. Para criação de Autoridades Subordinadas, a requisição de certificado, exportada na etapa 3, deve ser entregue a uma Autoridade Certificadora, para que ela emita o certificado digital. É importante destacar que a AC emissora pode alterar as informações da requisição de certificado da Autoridade, da forma que ela desejar. Portanto, o certificado resultante da emissão, pode possuir dados diferentes dos informados na na Etapa 3. Ao fim do processo, o certificado digital deve ser entregue ao operador responsável pela criação da autoridade.

Para a criação de uma Autoridade Certificadora Raiz, a requisição de certificado deve ser usada pelo próprio Módulo Gerenciador, para preencher a estrutura de um certificado digital, que será assinado com a chave privada criada na Etapa 1. Todos os campos obrigatórios do certificado digital, que não estão presentes na requisição de certificado, devem ser preenchidos pelo Módulo Gerenciador (sendo alguns deles fornecidos pelo Operador). Ao fim desse processo, o Espaço de Armazenamento da AC será alterado, pois o certificado é considerado um certificado emitido pela própria AC. Ou seja, o certificado deve possuir um número serial único no contexto da AC (causando a atualização dessa informação do Espaço de AC), e deve ser inserido no conjunto de certificados emitidos da AC.

**Etapa 5 (Armazenamento do Certificado)** O armazenamento do certificado

digital da Autoridade, consiste no armazenamento do certificado digital no Espaço de Armazenamento da Autoridade. Para a criação de Autoridades Subordinadas, o operador deve importar o certificado digital para o Módulo Gerenciador, que deve verificar se a chave pública, do certificado, corresponde à chave pública da requisição armazenada. Caso não corresponda, a criação não é efetivada. Se corresponder, o Módulo Gerenciador deve apresentar as informações alteradas no certificado, em relação a requisição de certificado. Então, fica a cargo do Operador decidir se aceita o certificado para armazená-lo no Espaço de Armazenamento da Autoridade. No caso de criação de uma Autoridades Certificadoras Raízes, o certificado deve ser armazenado automaticamente, após a Etapa 4.

### 5.3.5 Configuração do Intervalo de Emissão Automática de LCR

O Módulo Gerenciador de Autoridades Certificadoras deve fornecer funções para o Operador configurar o intervalo de tempo entre as emissões de listas de certificados revogados da sua Autoridade Certificadora. Esse intervalo é usado pelos Módulos Servidores de AC, que são responsáveis por emitir periodicamente uma nova lista de certificados revogados, conforme o intervalo determinado. Recomenda-se que a representação do intervalo seja em segundos, para que possua uma faixa de representação suficientemente grande e granular.

**Definição 25 (Intervalo de Emissão de LCR)** O intervalo de emissão de LCR é um número $t$, tal que $t \in \mathbb{N}^*$, que representa o intervalo em segundos entre as emissões automáticas de LCR.

Após a criação de uma Autoridade Certificadora, é preciso emitir uma LCR inicial, vazia (ADAMS et al., 2005). Como o campo *nextUpdate* da LCR é obrigatório (COOPER et al., 2008), o Operador deve definir, nesse momento, um valor para o intervalo de emissão automática de LCR. Esse valor será somado com a data de emissão da LCR inicial, e o resultado será atribuído ao seu campo *nextUpdate*.

Quando a Autoridade Certificadora já estiver em operação e o valor do intervalo ter que ser alterado, o Operador deverá escolher entre: utilizar o novo intervalo na próxima emissão de LCR (prevista pela última LCR emitida), ou então, emitir uma LCR imediatamente com o valor do novo intervalo. Essa escolha é importante, pois, caso o operador escolha a segunda opção, a LCR emitida irá anteceder a data prevista pela última LCR. Isso causa uma sobreposição de LCRs e pode causar problemas de sincroniza para

os Módulos Servidores de AC. As consequências relacionadas a essa escolha são discutidas na Seção 5.4

### 5.3.6 Gerenciamento de Servidores

O gerenciamento de servidores inclui a função de cadastramento de Módulos Servidores e as funções de autorização e desautorização, dos mesmos, para operarem uma Autoridade do Módulo Gerenciador. O Módulo Servidor é uma máquina conectada à uma rede de computadores, da qual o Módulo Gerenciador faz parte. Portanto, seu cadastramento consiste no armazenamento de uma URI (*Unique Resource Identifier*) e de um certificado digital, que identificam o servidor.

**Definição 26 (Servidor)** O servidor pode ser definido como uma tupla $S = (u, c, e)$, tal que, $u$ é a URI do servidor; $c$ é o certificado digital que identifica o servidor; e $e$ é o estado do servidor, sendo que $e \in \{habilitado, desabilitado\}$.

Após o cadastramento de um Módulo Servidor, o Operador pode autorizá-lo a operar a Autoridade, dando-o acesso aos recursos da Autoridade em questão. Para isso, ele precisa enviar uma mensagem para o Módulo Servidor, contendo uma chave de sessão para os recursos da Autoridade ($s$) e as informações de acesso aos recursos ($R$). Esse mensagem deve ser cifrada com a chave pública do Módulo Servidor ($f_c$) e assinada com a chave privada do Perfil de Operação da Autoridade ($f_d$). Essas operações criptográficas podem ser substituídas por uma conexão SSL de autenticação mútua.

Para que o servidor possa autenticar o Módulo Gerenciador, é preciso que ele importe o certificado do perfil de administração como uma âncora de confiança. Dessa forma, ele irá aceitar solicitações de todos os perfis de operação que fizerem parte do Módulo Gerenciador. O servidor também precisam manter o certificado do perfil de operação que solicitou o serviço, para que somente ele possa solicitar o encerramento das operações.

$$1. \quad G \quad \longrightarrow \quad S \quad : \quad \{\{s, R\}_{f_c}\}_{f_d}$$

Os Operadores também devem ser capazes de desautorizar um Módulo Servidor de operar uma determinada Autoridade. Para isso, o Módulo Gerenciador envia uma mensagem ao Módulo Servidor, contendo o identificador da Autoridade que deve ter a operação encerrada ($id$). Porém, a sessão do Módulo Servidor, com os recursos da Autoridade, deve ser destruída, garantido que ele não poderá mais utilizá-los.

$$1. \quad G \longrightarrow S : \{\{id\}_{f_c}\}_{f_d}$$

As funções de autorização e desautorização envolvem o banco de dados e o módulo de segurança criptográfico. Esses recursos são independentes do Módulo Gerenciador, portanto, o Perfil de Operação pode não ser capaz de autorizar o acesso a eles através do Módulo Gerenciador. Nesses casos, o operador deve executar as funções, especificas de autorização e desautorização, na aplicação de gerenciamento de banco de dados ou MSC, para então poder enviar ao servidor os dados de sessão para esses recursos. Esse assunto é discutido no Capítulo 6.

### 5.3.7 Gerenciamento dos Vínculos de Confiança

O vínculo de confiança entre uma AC e uma AR deve ser estabelecido através da troca de certificados digitais, que devem ser armazenados em seus respectivos espaços de armazenamento. Os certificados trocados podem ser os próprios certificados das autoridades. Porém, no caso de Autoridades Certificadoras, essa prática não é recomendada, pois resultaria no uso da sua chave privada para outro fim, além da emissão de certificados digitais e LCRs. Portanto, é recomendado que as Autoridades Certificadoras utilizem um par de chaves secundário relacionado com um certificado digital que a identifique, chamado de Certificado de Transporte.

O Módulo Gerenciador de AC faz o cadastramento de AR confiável salvando o certificado da Autoridade de Registro em uma lista de ARs confiáveis. Dessa forma, ao receber uma solicitação de uma AR, que deve estar assinada, a AC verifica se a assinatura da mesma pertence a uma Autoridade de Registro cadastrada como confiável.

No caso do Módulo Gerenciador de AR, é necessário que, além do certificado de transporte da Autoridade Certificadora, seja cadastrado uma URI para um servidor da AC em questão. Isso é necessário pois a AR sempre será responsável por iniciar a comunicação com a AC.

**Definição 27 (Lista de ARs Confiáveis)** Uma lista de Autoridades de Registro confiáveis consiste em um conjunto de certificados digitais que identificam as Autoridades de Registro confiáveis.

**Definição 28 (Lista de ACs Confiáveis)** Uma lista de Autoridades Certifica-

doras confiáveis consiste em um conjunto de tuplas $V = (c,u)$, tal que $c$ é o certificado de transporte da Autoridade Certificadora vinculada e $u$ uma URI para um Módulo Servidor responsável por operar a AC.

### 5.3.8 Gerenciamento dos Agentes de Registro e Instalações Técnicas

O Módulo Gerenciador de AR deve prover funções para o cadastramento, autorização, desautorização e associação de Agentes de Registro e Instalações Técnicas. O cadastro de agente de registro consiste no armazenamento de informações que podem ser usadas para identificá-lo e autenticá-lo. Recomenda-se que essa identificação seja feita através de certificados digitais com valor legal. Assim, os agentes de registro podem ser responsabilizados legalmente pelo uso das suas chaves privadas.

O cadastro de Instalação Técnica depende das tecnologias escolhidas pela Autoridade de Registro para identificá-la. Recomenda-se que essas identificações incluam, pelo menos, o computador de onde o Agente de Registro pode acessar o Módulo AGR. Porém, também podem ser aplicadas restrições relativas a posição geográfica, através de tecnologias de posicionamento global, como o GPS (Global Positioning System); e restrições de acesso à rede, através de VPN e *firewall*, por exemplo. Pela variedade de soluções, os dados relativos à identificação da Instalação Técnica são deixados a cargo da instituição que for desenvolver o SGC.

Além do cadastramento de Agente de Registro e Instalação Técnica, o Módulo Gerenciador de AR deve permitir que essas entidades sejam associadas. Cada Agente de Registro deve ser vinculado a uma ou mais Instalações Técnicas, das quais ele poderá acessar o Módulo AGR, sendo que cada Instalação Técnica pode ser associada a mais de um Agente de Registro.

**Definição 29 (Agente de Registro)** O agente de registro pode ser representado como uma tupla $A = (c,I)$, tal que $c$ é um certificado digital que o identifica e $I$ é um conjunto de Instalações Técnicas nas quais ele pode trabalhar.

**Definição 30 (Instalação Técnica)** Uma instalação técnica é definida pelos dados que identificam um perímetro físico-lógico de onde o Agente de Registro pode acessar o Módulo AGR.

## 5.4 MÓDULOS SERVIDORES

Os Módulos Servidores de AC e de AR e o Módulo Público e o Módulo AGR, interagem para compor as cerimônias de emissão de certificado digital, revogação de certificado digital e emissão de lista de certificados revogados. Sendo que uma cerimônia é como um protocolo, porém, ela também especifica as interações de máquinas com humanos e de humanos com humanos (ELLISON, 2007).

A seguir, em cada seção, uma dessas cerimônias será apresentada, junto com as especificações das funções dos módulos que a compõem. Essas cerimônias são recomendações que, portanto, podem ser implementadas de forma diferente. Porém, há uma razão para elas serem propostas da forma que estão, principalmente a cerimônia de emissão de certificado digital.

Nos sistemas avaliados, o Usuário Final tem que gerar seu par de chaves na primeira interação com AR, antes mesmo de agendar o encontro com o Agente de Registro. Dessa forma, o Usuário Final acaba sendo obrigado a definir uma senha (PIN) de proteção para sua chave privada, seja no próprio computador ou em um *smartcard*. Como a emissão do certificado acaba acontecendo após a aprovação dos AGR, o Usuário Final corre o risco de esquecer o seu PIN, ou até mesmo a máquina onde a chave está armazenada, antes da emissão do certificado. Isso faz com que ele não seja capaz de instalar o certificado digital para uso, acarretando na sua revogação.

A cerimônia proposta na Seção 5.4.1, tenta aproximar o momento em que a chave privada é gerada do momento em que o Certificado Digital é emitido. Assim, evita-se que o usuário se esqueça do seu PIN antes mesmo de instalar o certificado digital em sua máquina. Após a instalação do certificado, espera-se que o Usuário Final passe a usá-lo, reduzindo a possibilidade de esquecimento do seu PIN.

Vale lembrar que as partes da cerimônia que envolvem a comunicação entre o Módulo Público e o Servidor de AR, e entre o Servidor de AR e o Servidor de AC, são todas especificadas pelos protocolos de gerenciamento de certificados digitais. Portanto, não são de autoria deste trabalho. As trocas de mensagens são apresentadas apenas para facilitar o acompanhamento da descrição da cerimônia.

### 5.4.1 Emissão de Certificado Digital

O cerimônia de solicitação de certificado digital se inicia com um Usuário Final ($U$) selecionando uma Autoridade Certificadora da qual deseja obter um certificado digital e uma Autoridade de Registro associada a

ela. A forma de apresentação da AC e suas ARs para o usuário final está fora do escopo deste trabalho, porém, o Usuário Final deve ser capaz de acessar a aplicação do Módulo Público da AR escolhida ($M_{pub}$), através de um dos servidores da AR ($S_{ar}$). Um ponto importante, nesse momento, é que o Usuário Final deve ser capaz de autenticar a aplicação do Módulo Público, para garantir que ela pertence à Autoridade de Registro que ele escolheu.

$$1. \quad S_{ar} \quad \longrightarrow \quad U \quad : \quad \{M_{pub}\}_{ar}$$

Através do Módulo Público, o Usuário Final inicia o processo de solicitação de certificado digital, registrando seu interesse em obter um certificado digital emitido pela Autoridade Certificadora Escolhida ($ca$). Nesse momento, o Módulo Público pode solicitar que o Usuário Final informe dados para identificá-lo ($U_id$), como um nome e um número de documento. Esses dados serão enviados para o Servidor de AR e armazenados em seu Espaço de Armazenamento.

$$2. \quad M_{pub} \quad \longrightarrow \quad S_{ar} \quad : \quad \{U_{id}, ca\}$$

Completado seu registro inicial, o Usuário Final deve agendar uma data e hora para se apresentar em uma das instalações técnicas da AR. O agendamento não precisa ser feito, necessariamente, pelo Módulo Público, podendo ser feito por outras vias (e.g., telefone). Nesse momento, o Usuário Final pode ser instruído sobre os documentos necessários que ele tem que levar à instalação técnica para ter sua solicitação de certificado aprovada.

No dia e hora agendados, o Usuário Final se apresenta na instalação técnica, levando os documentos necessários para sua identificação e autenticação. Então, um Agente de Registro ($agr$), devidamente autorizado pela AR à acessar o Módulo AGR ($M_{agr}$) através daquela Instalação Técnica ($it$), busca a solicitação de certificado digital com as informações do Usuário Final. Então, o servidor da AR retorna uma lista contendo todas as solicitações de certificado do usuário informado ($S$).

$$3. \quad M_{agr} \quad \longrightarrow \quad S_{ar} \quad : \quad \{U_{id}\}_{agr+it}$$
$$4. \quad S_{ar} \quad \longrightarrow \quad M_{agr} \quad : \quad \{S\}_{ar}$$

Existindo uma solicitação, o Agente de Registro autentica o Usuário Final, conferindo seus documentos. Após a autenticação, o Agente de Registro preenche um formulário, gerado pelo Módulo AGR, com os dados a serem inseridos no certificado digital do Usuário Final. Então, envia esses dados para o Servidor da AR, junto com um identificado de aprovação do

certificado (‘*ok*‘). O Servidor da AR armazena as informações do Usuário Final e a aprovação do Agente de Registro.

$$5. \quad M_{agr} \longrightarrow S_{ar} : \{U_{inf}, \text{‘}ok\text{‘}\}_{agr+it}$$

Uma vez aprovado pelo primeiro Agente de Registro, os documentos são repassados para outros Agentes de Registro que devem realizar a validação das informações pré-aprovadas, com os documentos do Usuário Final. Esse processo se repete até atingir o número mínimo de aprovações necessárias para emitir o certificado digital. Com todas as aprovações, o Módulo AGR gera um código de autenticação, associado a solicitação de certificado, e o entrega para o Usuário Final. O Servidor de AR armazena o *hash* do código de autenticação, para poder autenticá-lo posteriormente.

$$6. \quad S_{ar} \longrightarrow M_{agr} : \{id, Au\}_{ar}$$
$$7. \quad agr \longrightarrow U : \{id, Au\}$$

Em seu ambiente, o Usuário Final busca pela sua solicitação de certificado, através do Módulo Público, informando seu código de autenticação. O Módulo Público solicita ao Servidor da AR as informações pré-aprovadas da solicitação feita pelo Usuário Final. O Servidor da AR retorna as informações e o Módulo Público as usa para gerar uma requisição de certificado digital para o Usuário Final. Durante esse processo, um par de chaves assimétricas é gerada no ambiente do Usuário Final, sendo a chave pública inserida na requisição de certificado e a chave privada usada para assinar essa requisição.

$$8. \quad M_{pub} \longrightarrow S_{ar} : \{id\}_{A_u}$$
$$9. \quad S_{ar} \longrightarrow M_{pub} : \{U_{inf}\}_{ar}$$

O Módulo Público envia a requisição de certificado ($R_{cert}$) para o Servidor da AR, que, por sua vez, solicita a emissão do certificado digital para o Servidor da AC ($S_{ac}$), enviando a requisição de certificado do Usuário Final. O Servidor de AC realiza a emissão do certificado digital, salva o certificado no seu espaço de armazenamento, e envia o certificado emitido de volta para o Servidor da AR, que o envia para o Módulo Público. Então, o Módulo Público instala o certificado digital do Usuário Final em sua máquina.

$$
\begin{array}{llcll}
10. & M_{pub} & \longrightarrow & S_{ar} & : \{R_{cert}\}_{A_u} \\
11. & S_{ar} & \longrightarrow & S_{ac} & : \{R_{cert}\}_{ar} \\
12. & S_{ac} & \longrightarrow & S_{ar} & : \{C\}_{ac} \\
13. & S_{ar} & \longrightarrow & M_{pub} & : \{C\}_{ar}
\end{array}
$$

### 5.4.2 Revogação de Certificado Digital

A cerimônia de revogação de certificado digital pode ocorrer de duas formas. A primeira ocorre através de uma solicitação feita pelo Usuário Final, através do Módulo Público, para a Autoridade de Registro revogar seu certificado. Para isso, o usuário precisa se autenticar como responsável pelo certificado digital. Ele pode fazer isso através de uma mensagem enviada para o Servidor da AR, contendo o código de autenticação que ele recebeu para emitir seu certificado ou uma assinatura digital feita com a chave privada do certificado que ele deseja revogar. O Servidor da AR irá verificar a autenticidade da mensagem e enviar a solicitação para o Servidor da Autoridade Certificadora.

$$
\begin{array}{llcll}
1. & M_{pub} & \longrightarrow & S_{ar} & : \{id\}_{A_u} \\
 & & \text{ou} & & \\
1. & M_{pub} & \longrightarrow & S_{ar} & : \{C\}_{usr}
\end{array}
$$

Quando o requerente não possuir nem o código de autenticação ou a chave privada do certificado que deseja revogar, ele deve fazer uma solicitação formal para a AR. Para isso, o requerente agenda uma data e hora para se apresentar em uma Instalação Técnica da AR. Lá, um Agente de Registro, através do Módulo AGR, busca as informações e requisições de certificados digitais do Usuário Final ($U_{inf}$ e $R$), enviando um identificador único do usuário ($U_{id}$) ao Servidor da AR.

$$
\begin{array}{llcll}
1. & M_{agr} & \longrightarrow & S_{ar} & : \{U_{id}\}_{agr+it} \\
2. & S_{ar} & \longrightarrow & M_{agr} & : \{U_{inf}, R\}_{ar}
\end{array}
$$

Então, o Agente de Registro autentica o Usuário Final, comparando seus documentos com as informações retornadas pelo Servidor da AR. Após a autenticação, caso exista mais de uma requisição de certificado aprovada, o Usuário Final terá que informar o Agente de Registro qual das requisições se refere ao certificado que ele deseja revogar. Então, o Agente de Registro aprova a revogação daquele certificado ($r$). Sendo que pode ser necessária a aprovação de mais de um Agente de Registro.

$$3. \quad M_{agr} \longrightarrow S_{ar} : \{S_{rev}, \text{`}ok\text{`}\}_{agr+it}$$

Aprovada a solicitação, o Servidor da AR envia uma requisição de revogação do certificado digital para o servidor de AC. O Servidor de AC revoga o certificado e envia o resultado da revogação de volta para o Servidor da AR, e esse para o Módulo AGR. O Agente de Registro, então, entrega ao requerente um comprovante da revogação do seu certificado digital.

$$4. \quad S_{ar} \longrightarrow S_{ac} : \{R_{rev}\}_{ar}$$
$$5. \quad S_{ac} \longrightarrow S_{ar} : \{\text{`}revogado\text{`}\}_{ac}$$
$$6. \quad S_{ar} \longrightarrow M_{agr} : \{\text{`}revogado\text{`}\}_{ar}$$

### 5.4.3 Emissão Automática de LCR

O Módulo Servidor de AC tem a responsabilidade de emitir uma nova lista de certificado revogados periodicamente, conforme o Intervalo de Emissão de LCR que é definido pelo Módulo Gerenciador (presente no Espaço de Armazenamento da AC). Para executar essa função, o Módulo Servidor de AC deve se inicializar através de uma consulta ao Espaço de Armazenamento da AC, onde irá buscar a última LCR emitida, recuperar a data presente no campo *nextUpdate* e agendar a emissão da LCR para essa data.

Na data prevista, o Módulo Servidor de AC deve consultar o Espaço de Armazenamento de AC, buscar a última LCR emitida, recuperar a data presente em seu campo *nextUpdate* e verificar se é a data atual. Se for a data atual, o Módulo Servidor de AC emite uma nova LCR com o valor *nextUpdate* igual a data atual mais o valor do Intervalo de Emissão de LCR que se encontra no Espaço de Armazenamento da AC. Se for uma data futura, significa que a nova LCR já foi emitida por outro Módulo Servidor. Portanto, ele não deve emitir a LCR e deve agendar a nova data prevista para emissão.

Quando o Módulo Gerenciador da AC, por alguma razão, alterar o intervalo de emissão de LCR, ele irá decidir entre: emitir uma nova LCR imediatamente; ou esperar a data prevista para emissão da próxima LCR. Se for decidido por esperar a data prevista pela última LCR, o procedimento de emissão automática de LCR não será afetado. O Módulo Gerenciador irá alterar o valor do Intervalo de Emissão de LCR no Espaço de Armazenamento da AC, e esse será usado pelo Módulo Servidor de AC quando ele for emitir a próxima LCR, conforme agendado.

Porém, caso seja escolhido realizar a emissão imediata da LCR, o Módulo Servidor de AC não será capaz de prever essa emissão, a não ser que rea-

lize consultas regulares ao Espaço de Armazenamento de AC. Para evitar que isso seja necessário, recomenda-se que o próprio Módulo Gerenciador realize a emissão dessa LCR. Dessa forma, ele atualizará o Espaço de Armazenamento da AC com a nova LCR emitida e com o novo Intervalo de Emissão de LCR.

É importante lembrar que, como recomendado pela RFC-5280, os emissores de LCR devem emitir LCRs com a data presente no campo *nextUpdate* igual ou maior do que as datas dos mesmos campos de todas as LCR anteriores (COOPER et al., 2008). Nesse contexto, essa recomendação impede que a nova LCR, emitida pelo Módulo Gerenciador, possua o campo *nextUpdate* com o valor menor do que a data agendada pelos Módulos Servidores. Assim, quando o Módulo Servidor tentar emitir a LCR, ele verificará que uma LCR já foi emitida (pelo Módulo Gerenciador) e agendará a emissão para a próxima data prevista.

## 5.5 CONCLUSÃO

Neste capítulo, foi proposto um modelo para sistemas gerenciadores de certificados digitais baseado no modelo de gerenciamento de certificados digitais apresentado no Capítulo 4. Esse modelo de SGC apresenta soluções que permitem a emissão distribuída e em larga escala de certificados digitais, tratando os módulos servidores de AC e AR como recursos que podem ser compartilhados entre várias ACs ou ARs, respectivamente.

Outras contribuições apresentadas neste capítulo incluem: o protocolo de autenticação de usuário, que reduz a probabilidade de corrupção do operadores do sistema; a geração e recuperação de backups, que garante que apenas os responsáveis pelo backup possam gerá-los e recuperá-los; e o protocolo de emissão de certificado digital, que reduz a possibilidade de revogação de certificados devido a erros cometidos pelo usuário final (requerente de certificado).

# 6 DISCUSSÃO

## 6.1 INTRODUÇÃO

Neste capítulo, são discutidas as contribuições deste trabalho e como elas solucionam os problemas da emissão distribuída e em larga escala de certificados digitais. Isso é feito através de uma dissertação sobre como os problemas foram atacados para que fossem atingidos os objetivos listados na Introdução deste trabalho. Além disso, como parte da análise, foi implementado um protótipo, para avaliar a viabilidade do modelo proposto no Capítulo 5.

## 6.2 CONTRIBUIÇÕES DO NOVO MODELO

O Modelo de SGC, proposto no Capítulo 3, apresenta uma extensão do modelo de entidades da ICP *X.509*. Esse modelo inclui uma nova entidade, chamada de Agente de Registro, responsável pela aprovação e rejeição das requisições feitas pelos Usuários Finais às Autoridades de Registro. Além disso, o modelo proposto detalha as aplicações, máquinas e pessoas que constituem os sistemas utilizados pelas entidades do modelo. Esse detalhamento trás diversos benefícios para os processos de desenvolvimento de sistemas gerenciadores de certificados digitais, sendo os principais apresentados nas seções a seguir.

### 6.2.1 Separação AC e AR

A separação das funções de Autoridade Certificadora e Autoridade de Registro já é prevista nos modelos propostos pelas RFCs da ICP *X.509* (IETF, 2013). Porém, nesses modelos é dada pouca ênfase à AR. No modelo proposto pelo CMP, a AR se encontra no mesmo ambiente da AC e não possui funções definidas especificamente para ela. Ela é tratada como um Usuário Final perante a AC e como uma AC perante o Usuário Final. Ou seja, a AR é apresentada apenas um gargalo de requisições que redireciona as requisições dos Usuários Finais para a Autoridade Certificadora. Essa abstração para a entidade AR pode até simplificar o protocolo de gerenciamento de certificados digitais. Porém, quando usada como modelo de desenvolvimento para Sistemas Gerenciadores de Certificados Digitais, pode causar problemas de

interpretação que reduzem a coesão e aumentam o acoplamento dos sistemas implementados.

Dos SGCs avaliados, parte implementa as funções de AC e AR e outra parte implementa apenas as funções de AC. Dos SGCs que implementam as funções de AC e AR, todos implementam as funções na mesma aplicação. Mas, mesmo que a separação dessas funções seja clara para os usuários do software, essa prática pode ser nociva para o SGC, já que brechas de segurança nas funções da AR podem levar ao comprometimento da AC que compartilha a mesma aplicação. No caso dos SGCs que implementam apenas as funções de AC, notou-se que todos implementam, também, funções de verificação e aprovação de requisições de certificado digital. Ou seja, parte das funções previstas para a AR são embutidas nas funções da AC.

Para sanar esses problemas, o modelo de SGC proposto neste trabalho apresenta uma separação mais nítida entre as entidades AC e AR, definindo ambientes e módulos separados para cada uma dessas entidades. Além disso, a remoção da opcionalidade do uso de Autoridades de Registro garante a coesão dessas entidades, pois remove a possibilidade de existirem sistemas de AC que implementam parcialmente as funções de AR. De forma geral, esse modelo proporciona uma arquitetura mais completa e coesa do que o modelo da ICP *X.509*.

No contexto das ICPs Governamentais, a separação de AC e AR é uma prática compatível com seus modelos de organização. Por exemplo, a instituição responsável pelo cadastramento de pessoas físicas e jurídicas (CPFs e CNPJs) no Brasil é a Receita Federal. Porém, os processos burocráticos de identificação e autenticação dessas pessoas são feitos por outras instituições. Assim, as entidades responsáveis pela parte burocrática de identificação e autenticação podem assumir o papel de Autoridades de Registro, enquanto o documento (certificado digital) é emitido por uma AC da Receita Federal.

### 6.2.2 Inclusão do Agente de Registro

No modelo da ICP *X.509*, apenas o Usuário Final se comunica com a AR. Porém, são determinadas funções para AR que não são especificadas pelo modelo. Essas funções incluem a existência de um usuário responsável por aprovar e rejeitar as requisições feitas pelos Usuários Finais está presente na maioria dos SGCs Online avaliados. No modelo proposto, esse usuário é incluído com uma entidade atuante na gerência de certificados digitais e separada da AR, chamada de Agente de Registro.

A inclusão do Agente de Registro permitiu separar as funções da AR, previstas pelos CMP, das funções dos usuários que operam as ARs. De certa

forma, essa separação já existia nos SGCs Avaliados, pois, na sua maioria, os SGCs fornecem uma interface web para esses usuários. Porém, essas interfaces misturavam funções de configuração da AR com as funções de aprovação e rejeição de requisições, além de serem acopladas à AR. A definição clara das funções do responsável pela aprovação ou rejeição de uma requisição de certificado digital é uma ferramenta útil para a criação de evidências

No contexto das ICPs Governamentais, a definição clara das funções do Agente de Registro são compatíveis com os modelos de governos semelhantes ao brasileiro. Os funcionários públicos que assumem cargos de confiança e possuem fé pública em suas rubricas são perfeitamente interpretáveis como Agentes de Registro.

### 6.2.3 Identificação dos Módulos e Recursos

Além da separação entre AC e AR, este trabalho propõe um modelo de SGC mais detalhado, que separa em módulos distintos as funções de gerenciamento e de serviços das Autoridades Certificadoras e de Registro. Também são identificados os recursos usados por essas Autoridades durante a execução das suas funções, são eles: o Espaço de AC, o Espaço de AR e o Provedor Criptográfico.

Esse detalhamento reduz a abstração feita pelas RFCs sobre as entidades da ICP, que apenas especificam as funções de serviço dessas autoridades. Através desses módulos e recursos foi possível especificar um conjunto maior de funções que fazem parte dos casos de uso da maioria dos Sistema Gerenciador de Certificados Digitais.

## 6.3 CONTRIBUIÇÕES DO SISTEMA ONLINE

### 6.3.1 Protótipo

Nesta seção é apresentada uma avaliação sobre o Sistema Online proposto no capítulo anterior. Para auxiliar nessa avaliação, foi implementado um protótipo, que contempla a maioria das funções especificadas para o Sistema Online. O objetivo da sua implementação é avaliar a completude do modelo e as dificuldades em convertê-lo em uma aplicação através das tecnologias existentes para o âmbito de ICP.

O protótipo foi desenvolvido em duas linguagens de programação: C++ em conjunto com o *framework* QT (Digia, 2013), para os Módulos Gerenciadores de AC e AR; e Java Web em conjunto com os servidores de apli-

cação Tomcat e Glassfish, para os Módulos Servidores de AC e AR, o Módulo Público e o Módulo AGR. Para simular a execução dos módulos em servidores distintos, foram utilizadas máquinas virtuais com o sistema operacional Linux *Red Hat* 5.5. Uma para cada módulo e banco de dados. O MSC utilizado foi um MSC de testes da Kryptus, modelo AHX4 ASI-HSM (Kryptus, 2013), que não possui restrição de acesso para uma única máquina e pode ser conectado a uma rede de computadores.

O protocolo de gerenciamento de certificados digitais utilizado foi o CMC, disponibilizado pela Mozzila (Mozilla, 2013), sendo o transporte das mensagens realizado através do protocolo *RESTful* (Oracle, 2013a), que permite um cliente solicitar serviços de um servidor através de requisições HTTP (Hyper Text Transfer Protocol). Para o balanceamento de carga, foi utilizada a aplicação *crossroads* (E-tunity, 2013), disponível no repositório de aplicações do Ubuntu.

Os Espaços de Armazenamento de AC e de AR foram implementados no formato de bancos de dados relacionais, sendo utilizado o SGBD (Sistemas Gerenciadores de Banco de Dados) MySQL. Já os provedores criptográficos são baseados nas bibliotecas criptográficas *Bouncy Castle* (HOOK, 2010) e *Libcryptosec* (LabSEC, 2010), sendo a comunicação com o MSC feita através da API *Engine OpenSSL* (OpenSSL, 2013).

A biblioteca *Libcryptosec* foi desenvolvida pelo LabSEC e é uma abstração da biblioteca *Libcrypto* do OpenSSL para o paradigma de orientação a objetos em C++. Como a biblioteca criptográfica *Bouncy Castle* é implementada em Java e não possui suporte à comunicação com MSC através da API *OpenSSL Engine*, foi implementada uma interface em Java para as funções da *Libcryptosec*, através da tecnologia JNI (Java Native Interface) (Oracle, 2013b).

Para a comunicação com *smartcards* foi utilizada a API PKCS#11 (RSA Laboratories, 2009). Nos Módulos implementados em C++ foi utilizada a biblioteca *Libp11* (JELLINGHAUS et al., 2013), abstraída pela *Libcryptosec*; enquanto nos Módulos implementados em Java, foi utilizado o provedor PKCS#11 da IAIK (*Institute for Applied Information Processing and Communication*) (IAIK, 2013).

### 6.3.2 Distribuição de Servidores

Através do Sistema Online proposto, a separação dos módulos de gerência e serviço foi estendida. Os Módulos Servidores passaram a ser tratados com recursos que podem ser compartilhados entre vários Módulos Gerenciadores. Essa separação permite que a tecnologia utilizada nesses Módulos

Servidores seja padronizada em sistemas embarcados ou *appliances* dedicados, da mesma forma que é feito com as urnas eletrônicas brasileiras. A padronização da tecnologia de software e hardware facilita a instalação da solução, reduz custos, garante compatibilidade entre aplicações e facilita a homologação do sistema.

Na figura 26, são apresentados três cenários de testes onde os Módulos Servidores são configurados para alcançarem desempenho (através de balanceamento de carga), eficiência, disponibilidade (através de redundância) e distribuição geográfica. Em todos exemplos são usados Módulos do Ambiente de AC, mas todas as configurações podem ser usadas em um Ambiente de AR. Cada uma das configurações possui duas Autoridades Certificadoras gerenciadas por um Módulo Gerenciador de AC.

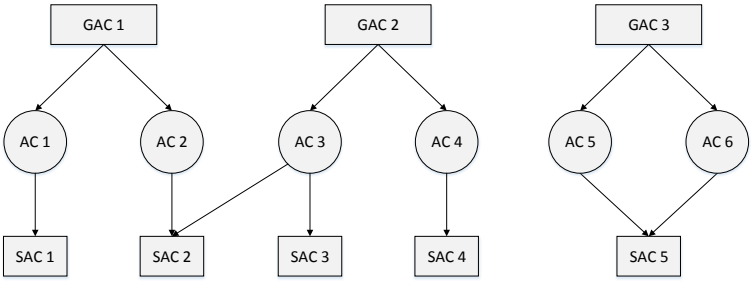


Figura 26 – Exemplos de Configuração dos Módulos

A configuração da esquerda ilustra um SGAC que distribui cada uma de suas ACs em servidores diferentes. Esse tipo de configuração é útil para ICPs que distribuem diversas ACs em uma grande área geográfica, mas que ao mesmo tempo precisam manter um controle centralizado de todas elas. Também pode ser usado para evitar a redução da qualidade de serviço que seria causada por todas ACs estarem operando em um mesmo servidor. A configuração da direita mostra um SGAC que disponibilizas suas ACs em apenas um servidor (SAC 5). Esse tipo de configuração garante uma maior eficiência, pois usa menos recursos (SAC) para manter mais de uma AC operando. Entidades que controlam muitas ACs, mas possuem poucos usuários, podem usar essa configuração pela sua simplicidade e custo reduzido.

A configuração do centro apresenta uma configuração híbrida. Ela possui uma de suas ACs em um servidor dedicado e a outra em dois servidores diferentes, sendo um servidor compartilhado com uma AC do SGAC 1. A distribuição de uma AC em mais de uma servidor permite que esses servidores sejam configurados em um equipamento de balanceamento de carga

ou de redundância contra falhas, garantindo performance e disponibilidade para as funções da AC. Outro uso de múltiplos servidores é nos casos onde a distância geográfica pode prejudicar a qualidade do serviço. Através dessa configuração, é possível distribuir servidores da mesma AC em uma grande área geográfica.

Essas e outras configurações para os Módulos do SGC estão a favor das necessidades de uma ICP Governamental. Elas possibilitam que os serviços das Autoridades sejam distribuídos por todo território em que a ICP atua, atendendo as as necessidades especificas de cada região. E mesmo com essa distribuição, se mantém centralizadas, no Módulo Gerenciador, as funções de configuração, controle e auditoria das Autoridades e Servidores. Outro ponto positivo é que todas as configuração são escaláveis. Conforme as demandas das ACs aumentam, ou novas ACs são criadas, novos servidores podem ser instalados e configurados pelos Módulos Gerenciadores.

### 6.3.3 Compartilhamento dos Recursos

O maior desafio para a utilização do Modelo de Implementação proposto neste trabalho, é o compartilhamento dos recursos das Autoridades entre os Módulos Servidores que a operam. Esse compartilhamento não é trivial, pois o comprometimento do Espaço de Armazenamento da Autoridade ou do seu Provedor Criptográfico pode acarretar na revogação do seu certificado digital, o que pode ter consequências graves na Infraestrutura de Chaves Públicas em que atua.

O Espaço de Armazenamento da Autoridade pode ser facilmente implementado em um banco de dado relacional, que possui diversas soluções de compartilhamento e controle de acesso. Porém, dar acesso irrestrito de leitura e escrita aos Módulos Servidores pode não ser uma boa prática. Espera-se que esses servidores não venham a ser comprometidos, porém, caso sejam, é necessário proteger os dados da Autoridade. Para isso, sugere-se que o Espaço de Armazenamento de AC possua uma interface de acesso bem definida, que especifique apenas as funções necessárias para a operação da Autoridade.

Por exemplo, quando um Servidor de AC desejar emitir um certificado digital, ao invés de ele fazer uma consulta SQL ao banco de dados relacional, ele solicita à Interface de Armazenamento da AC todos os dados necessários para a emissão. A chamada dessa função faz com que a própria base de dados realize as consulta e atualize os campos necessários do Espaço de Armazenamento da AC. Além disso, com esse acesso controlado, é possível gerar *Logs* relativos que preservem as informações sobre quem solicitou determinado serviço em determinada data e hora. A proposta dessa interface ficou

fora do escopo deste trabalho, mas se beneficia da padronização estabelecida para os Espaços de Armazenamento propostos.

O compartilhamento de chaves criptográficas de uma Autoridade é um desafio ainda maior. A chave privada de uma autoridade é sua propriedade mais sensível, tanto que são armazenadas em Módulos de Segurança Criptográficos. Normalmente, os módulos de segurança criptográficos se associam com apenas uma máquina, através de uma porta *PCI*, *USB* ou de rede *Ethernet*. Essa restrição impede que um módulo de segurança criptográfico seja acessado, ao mesmo tempo, por mais de um computador. Portanto, a princípio, a solução de compartilhamento de chaves criptográficas consiste na replicação da chave em módulos de segurança criptográficos diferentes (Figura 27).

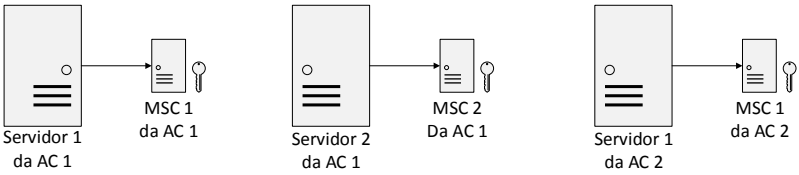


Figura 27 – Compartilhamento de Chaves Através de Replicação.

Porém, existem trabalhos que desencorajam essa prática. A replicação das chaves da Autoridade faz com que se perca a rastreabilidade das operações realizadas com ela. Além disso, se torna mais fácil perder o controle de cópias realizadas sobre cópias. Essa perda de controle é prejudicial pois, em casos de sinistro, a Autoridade terá dificuldade de identificar os responsáveis e reagir com ações que impeçam o comprometimento de toda Autoridade (SUTIL, 2011).

Para solucionar esse problema, é possível realizar um interfaceamento entre o MSC e os Módulos Servidores, através de um *middleware*, como ilustrado pela Figura 28. No mercado, já existem algumas soluções de MSCs remotos, baseados em *Nuvem*, como o serviço de *Cloud HSM* da *Amazon* (Amazon, 2013). Porém ainda são necessários estudos que verifiquem como esses modelos de acesso remoto se encaixam nos requisitos de segurança impostos pelos documentos que normalizam MSCs (FIPS, 2001) e pelos documentos que normalizam esses equipamentos na ICP (ITI, 2007b, 2007a).

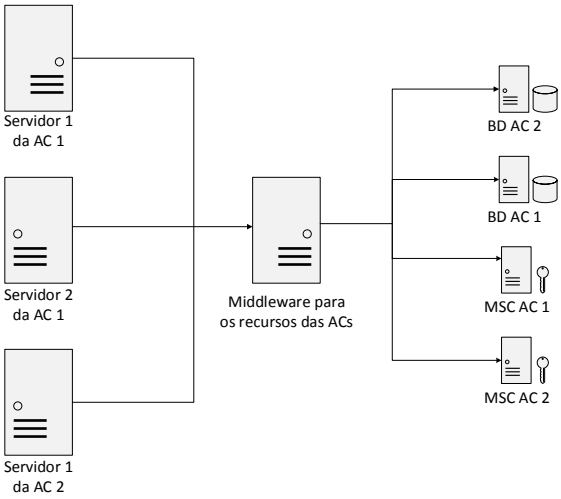


Figura 28 – Compartilhamento de Chaves Através de um *Middleware*.

### 6.3.4 Requisitos de Segurança e Usabilidade

No sistema online, os requisitos de segurança e usabilidade são propostos conforme as necessidades dos módulos e seus usuários. Por exemplo, os módulos gerenciadores e servidores de AC e AR possuem requisitos de segurança bem elevados, enquanto os módulos de Usuário Final e Agente de Registro apresentam requisitos de segurança de menor complexidade, porém com um nível de usabilidade mais elevado.

Nos módulos gerenciadores e servidores, os requisitos de segurança devem ser elevados pois esses módulos possuem um contato direto com os recursos sensíveis das ACs e ARs. E mesmo que esses níveis de segurança prejudiquem a usabilidade do sistema, é esperado que seus usuários sejam profissionais bem treinados, que têm os conhecimentos necessários para entender, operar e se responsabilizar pelo sistema. Além disso, esses usuários não possuem uma interação frequente com os seus módulos e, quando ocorre a interação, costumam-se seguir cerimônias pré-definidas que são acompanhadas por mais de uma testemunha.

O Módulo de Agente de Registro apresenta requisitos de segurança e usabilidade mais equilibrados, pois não há contato direto com o material sensível da AR e o módulo possui uma interação mais frequente com seus usuá-

rios. Dessa forma, as regras de controle de acesso podem ser mais brandas do que as regras de controle de acesso dos módulos gerenciadores e servidores. O Módulo Público, por sua vez, é acessado por Usuários Finais, que não possuem, obrigatoriamente, conhecimento ou treinamento para operá-lo. Dessa forma, o Módulo Público deve possuir níveis elevados de usabilidade e, consequentemente, mecanismos de segurança mais simplificados.

No sistema online proposto no Capítulo 5, os requisitos de segurança relacionados aos Módulos Gerenciadores e Servidores são cobertos através dos diversos mecanismos utilizado para autenticar usuários, servidores e backup. Todos esses mecanismos foram implementados e testados no protótipo, valendo destacar o controle de acesso ao espaço de armazenamento da AC ou AR. A segurança do banco de dados foi baseada em login e senha, dessa forma, cada perfil de usuário tinha seu login e senha cifrado com a chave pública do seu certificado.

A autenticação através de certificados digitais e a utilização de mais de um agente de registro para aprovar uma requisição de certificado se mostram mecanismos eficientes para esse propósito.

## 6.4 CONCLUSÃO

Através das análises realizadas neste capítulo, mostrou-se os benefícios trazidos pelos modelos propostos neste trabalho. Além disso, com a implementação do protótipo foi possível provar a viabilidade dos modelos propostos e destacar os desafios, não previstos anteriormente, relacionados com a emissão distribuída e em larga escala de certificados digitais. Como, por exemplo, o compartilhamento da chave privada da AC ou AR.

Através das contribuições discutidas neste capítulo, conclui-se que todos os objetivos deste trabalho foram alcançados. Os componentes da ICP foram apresentados de forma granular e todas suas funções foram especificadas; o uso correto de autoridades de registro, como entidades completamente independentes de AC; o agente de registro foi introduzido no modelo, como parte integrante do gerenciamento de certificados digitais; os recursos utilizados por ACs e ARs foram destacados e especificados, criando, dessa forma, o início de uma padronização para eles; foram especificados mecanismos de controle de acesso e backup que correspondem à natureza sensível de ICPs; foi especificado, também, um protocolo de emissão de certificado digital, com o objetivo de reduzir os erros do usuário final; e, por fim, com todas essas contribuições, foi proposto um sistema gerenciador de certificados digitais, capaz de emitir certificados digitais de forma distribuída e em larga escala.

# 7 CONSIDERAÇÕES FINAIS

Estre trabalho propôs dois modelos para Sistemas Gerenciadores de Certificados Digitais. O primeiro, apresentado no Capítulo 3, foi baseado nas análises sobre o modelo de gerenciamento de certificados digitais da ICP *X.509* e sobre as características de SGCs que implementam esse modelo. Essa análise destacou várias funções relacionadas à gerência de certificados digitais que não são tratadas pelo modelo *X.509*. Através dessas diferenças, o novo modelo de SGC foi proposto, com uma especificação mais detalhada dos componentes de *software* e *hardware* que integram o SGC.

Um dos objetivos deste trabalho era definir o uso correto de ARs, através de uma especificação mais enfática das suas funções. O uso de Autoridades de Registro é uma prática comum de diversas Autoridades Certificadoras. Porém, em muitos casos, a instituição responsável pela AC também é responsável pela AR. Dessa forma, muitos dos SGCs implementam as funções de AC e AR de forma acoplada. Esse acoplamento está associado à visão comercial sobre as Infraestruturas de Chaves Públicas. Como cada instituição possui suas próprias regras de negócio, tais regras acabam sendo implementadas no Sistema de AR. Isso faz com que a AR só possa prestar seus serviços para as ACs da mesma instituição.

Em Infraestruturas de Chaves Públicas de maior escala, o gerenciamento de certificados digitais pode envolver diversas instituições, sendo que muitas vezes essas instituições precisam interagir para realizar as funções da ICP. Dessa forma, o acoplamento entre os Sistemas de AC e AR é prejudicial para a ICP, pois restringe de diversas formas sua organização e pode ocasionar o desperdício de recursos. O novo modelo de SGC proposto, atinge o objetivo, especificando a AR como uma entidade obrigatória no Sistema Gerenciador de Certificados Digitais. Sendo suas funções apresentadas em módulos bem definidos com uma separação clara do Sistema de AC.

Além da especificação mais precisa para as Autoridades de Registro, este trabalho também propõe uma nova entidade para integrar o protocolo de gerenciamento de certificados digitais. Essa entidade é o Agente de Registro, responsável por atender o Usuário Final e validar suas requisições de emissão e revogação de certificados digitais.

Na ICP X.509, essas funções são atribuídas diretamente à AR. Porém, tais funções não possuem uma especificação precisa, de como devem ser executadas. Essa abstração ocorre devido as diversas regras de validação e aprovação de requisições que podem ser estipuladas pelas ACs e ARs. Porém, a especificação desse agente não deixa de ser importante, pois é definida a entidade responsável pela validação real da requisição de certificado digital.

Além disso, na maioria das vezes, as funções do Agente de Registro são assumidas por uma ou mais pessoas. Através das especificações das suas funções, é possível implementar mecanismos que geram evidências que permitem vincular a execução de uma determinada função com a pessoa que a executou. Funcionalidade muito importante para desestimular a corrupção dos Agentes de Registros e para auxiliar na resposta a possíveis ataques realizados contra a AR.

Também são especificados, nesse novo modelo de SGC, os recursos utilizados pelas ACs e ARs. Esses recursos incluem o provedor criptográfico e o espaços de armazenamento de AC e de AR. Entre esses dois recursos, destaca-se os espaços de armazenamento, que definem quais são os dados essenciais para a execução das funções de AC e AR. Através dessa definição, é possível estabelecer interfaces de acesso padronizado, da mesma forma que é feito com os Módulos de Segurança Criptográficos.

O segundo modelo se trata de um modelo de implementação para Sistemas Gerenciadores de Certificados Digitais Online. Entre suas propostas, destaca-se o diagrama de implantação, que estabelece a independência entre os Módulos de Gerência e de Serviço. Ele permite que os Servidores de AC e AR sejam tratados como recursos do SGC, configuráveis para a obtenção de performance, redundância, eficiência e distribuição geográfica dos serviços de AC e AR.

O modelo de implementação também propõe mecanismos de autenticação que atendem os rígidos requisitos de segurança das Infraestruturas de Chaves Públicas. Através da ICP interna proposta para o SGC, a divisão de funções entre os perfis de usuários propostos e o esquema de autenticação *M* de *N*, os Módulos de Gerenciamento do SGC se tornam resistentes à ataques externos e internos.

O backup é uma função presente em todos SGCs analisados, porém, cada um possui uma implementação própria. Os mais simples são apenas replicações de bases de dados, baseadas no sistema de arquivos do sistema operacional, protegidas pelo controle de acesso disponível pelo SGBD utilizado. Os sistemas mais complexos utilizam ferramentas externas ou internas, que fazem backup da base de dados do SGC e o cifram com uma chave simétrica, gerada durante a geração do backup.

Através dos mecanismos de controle de acesso e da padronização dos Espaços de Armazenamento de AC e AR, este trabalho propõe um sistema de backup seguro, que traz junto todas as características do sistema de autenticação proposto. O backup é protegido contra leitura e escrita através do ciframento de dados com a chave pública do perfil responsável pelo backup. Além disso, a origem do backup também é garantida, através da assinatura realizada pelo perfil que o gerou. Com esses mecanismos, o backup fica pro-

tegido contra ataques externos e se torna resistente à ataques internos, já que será necessário realizar a autenticação *M* de *N* para ter aceso ao conteúdo do backup.

Como muitas vezes o SGC gerencia mais de uma AC ou AR, pode-se tornar necessário que apenas uma dessas ACs ou ARs sejam exportadas para backup, seja por protocolos de segurança ou para transferência da entidade para outro sistema. Dessa forma, o backup específico de AC ou AR se mostra uma ferramenta importante para o SGC.

Por fim, o último objetivo deste trabalho era especificar uma cerimônia de emissão de certificados digitais mais amigável para o Usuário Final. Essa tarefa não é trivial, pois segurança e usabilidade são duas propriedades que costumam ser inversamente proporcionais (STRAUB, 2006). Devido aos inúmeros processos burocráticos e requisitos de segurança, a cerimônia de emissão de certificado não é muito maleável. Porém, algumas boas práticas podem auxiliar sua usabilidade para o Usuário Final.

Na cerimônia de emissão de certificado digital proposta no Capítulo 5, o Usuário Final interage, sozinho, com o SGC em apenas dois momentos: no início, fazendo a solicitação de emissão do seu certificado; e no final, durante a efetivação da emissão. Mesmo nesses dois momentos, o Usuário Final não precisa preencher fomulários extensos ou fazer escolhas que exigem conhecimento técnico sobre certificados digitais (e.g., algoritmo e tamanho da chave assimétrica, algoritmo de hash, etc.). Esses formulários e escolhas são feitos ao lado de um Agente de Registro, que, diferente do Usuário Final, é um usuário treinado e com conhecimento técnico sobre certificados digitais.

Além da redução das responsabilidades do Usuário Final, a cerimônia também faz com que a geração das suas chaves criptográficas ocorram logo antes da emissão do certificado digital. Esse detalhe é importante, pois, reduz a possibilidade do Usuário Final esquecer sua senha de acesso à chave privada, ou até mesmo a máquina onde ele gerou suas chaves. Como explicado no Capítulo 5, esses eram um dos principais motivos de revogação de certificados digitais de um dos SGCs analisados.

De forma geral, este trabalho contribui para qualquer processo de desenvolvimento de Sistemas Gerenciadores de Certificados Digitais. Porém, ele se destaca para os processos de desenvolvimento de SGCs usados em ICPs Governamentais, por apresentar um modelo que cumpre com diversos requisitos de segurança e que está preparado para ambientes de larga escala. Além disso, ele pode ser usado como base para a definição de documentos de homologação de Sistemas Gerenciadores de Certificados Digitais, como o MCT 11 da ICP-Brasil (ITI, 2010b, 2010a).

## 7.1 TRABALHOS FUTUROS

Apesar deste trabalho abordar diversos detalhes dos Sistemas Gerenciadores de Certificados Digitais, ainda é necessário produzir especificações que complementam o modelo proposto. As principais especificações incluem:

- **Interface de Acesso ao Espaço de Armazenamento:** a padronização dos dados a serem armazenados pelo SGC permite que uma interface de acesso seja especificada. Da mesma forma que os módulos de segurança criptográficos, o interfaceamento restringe a forma de acesso aos dados e permite gerar evidências de quem solicitou a leitura e escrita de determinados dados. Essas restrições trazem mais um nível de segurança para o SGC, garantindo a integridade do banco de dados e protegendo os dados contra leitura e escrita não autorizadas ou não definidas pelas funções de AC e AR.

  Por exemplo, na emissão do certificado digital, o Módulo Servidor de AC precisa recuperar dados do Espaço de Armazenamento de AC, como, por exemplo, o número serial que deve ser usado no certificado que será emitido. Além disso, o número serial precisa ser atualizado de forma correta e espera-se que, posteriormente, o certificado emitido seja armazenado no Espaço de Armazenamento da AC. Com a interface de acesso, é possível definir uma função que retorna todos os dados necessários para a emissão do certificado digital, e que já atualiza o número serial quando esses dados são solicitados. Além disso, a interface passa a esperar que o Módulo Servidor, que solicitou os dados, solicite o armazenamento do certificado digital com o número serial fornecido.

- **Compartilhamento do Provedor Criptográfico:** Como discutido no Capítulo 6, o provedor criptográfico precisará ser compartilhado entre os Módulos que compõem o Sistema de AC ou AR. Esse compartilhamento não é trivial e precisa cumprir com diversos requisitos de segurança. Estudos mais aprofundados sobre o compartilhamento de chaves precisam ser realizados, para que possa ser especificado um modelo compatível com o proposto neste trabalho. Alguns trabalhos já abordam esse tema (SUTIL, 2011), sendo alguns deles implementados e utilizados comercialmente, como o *Cloud HSM* da Amazon (Amazon, 2013).

- **Detalhamento de Protocolos:** O modelo proposto define dois novos protocolos, o de comunicação entre o Módulo Gerenciador e o Módulo Servidor (de AC ou AR), e o protocolo de comunicação entre o Agente de Registro e AR. Esses protocolos são especificados através

de uma notação formal e parte das suas mensagens são abstraídas, para que possam ser utilizadas em contextos diferentes. Porém, se mostra necessária uma especificação mais detalhada desses protocolos e suas mensagens, da mesma forma que os protocolos CMC e CMP são especificados.

- **Sistema de *Log* e Auditoria:** Apesar do modelo proposto neste trabalho definir a existência de um perfil de auditoria e de um Espaço de Armazenamento de Logs, ele não contribui extensivamente nesse contexto. A complexidade dos processos de auditoria e do armazenamento seguro de Logs extrapola os limites deste trabalho. Portanto, foram deixados como um trabalho futuro, que deverá ser realizado de forma especializada para esses problemas. O Laboratório de Segurança em Computação já abordou esse problema em outros projetos (MENTI, 2008), portanto já possui uma base de pesquisa inicial, que poderá ser usada na especificação completa de um sistema de Logs e Auditoria para Sistemas Gerenciadores de Certificados Digitais.

# REFERÊNCIAS

ADAMS, C. et al. **[RFC-4210] Internet X.509 Public Key Infrastructure - Certificate Management Protocol (CMP)**. IETF, set. 2005. RFC 4210 (Informational). (Request for Comments, 4210). Disponível em: <http://tools.ietf.org/html/rfc4210>.

Amazon. **AWS CloudHSM**. Amazon, 2013. Disponível em: <http://aws.amazon.com/cloudhsm/>.

BELLA, G.; COLES-KEMP, L. Layered analysis of security ceremonies. In: ____. **Information Security and Privacy Research**. [S.l.]: Springer, 2012. v. 376, p. 273–286. ISBN 978-3-642-30435-4.

CARLOS, M. C. et al. An updated threat model for security ceremonies. In: **Proceedings of the 28th Annual ACM Symposium on Applied Computing**. New York, NY, USA: ACM, 2013. (SAC '13), p. 1836–1843. ISBN 978-1-4503-1656-9. Disponível em: <http://doi.acm.org/10.1145/2480362.2480705>.

COOPER, D. et al. **[RFC-5280] Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile**. IETF, maio 2008. RFC 5280 (Informational). (Request for Comments, 5280). Disponível em: <http://tools.ietf.org/html/rfc5280>.

COSTA, C. B. M. **Melhorias de Usabilidade e Segurança para o SGCI**. Tese (Trabalho de Conclusão de Curso) — Universidade Federal de Santa Catarina, 2012. Disponível em: <https://projetos.inf.ufsc.br/arquivos_projetos/projeto_1220/TCC_CesarBarone_final.pdf>.

DIFFIE, W.; HELLMAN, M. New directions in cryptography. In: IEEE. **IEEE Transactions on Information Theory**. IEEE, 1976. v. 22, n. 6, p. 644–654. ISSN 00189448. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1055638>.

Digia. **QT Project**. Digia, 2013. Disponível em: <http://qt-project.org/>.

E-tunity. **Crossroads**. E-tunity, 2013. Disponível em: <http://crossroads.e-tunity.com/>.

ELLISON, C. Ceremony Design and Analysis. In: CITESEER. **Citeseer**. Cryptology ePrint Archive, Report 2007/399, 2007. v. 399, p. 1–17. Disponível em: <http://eprint.iacr.org/2007/399>.

FIPS. **Security Requirements for Cryptographic Modules**. [S.l.], maio 2001. viii + 61 p. (FIPS PUB, v. 140-2). Annex A: Approved Security Functions (19 May 2005); Annex B: Approved Protection Profiles (04 November 2004); Annex C: Approved Random Number Generators (31 January 2005); Annex D: Approved Key Establishment Techniques (30 June 2005). Supersedes FIPS PUB 140-1, 1994 January 11. Disponível em: <http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf>.

GARFINKEL, S. **PGP: pretty good privacy**. [S.l.]: O'reilly, 1995.

HOHNSTäDT, C. **X Certificate and key management**. maio 2012. Disponível em: <http://xca.sourceforge.net/>.

HOOK, D. **Bouncy Castle**. 2010. Disponível em: <http://www.bouncycastle.org/>.

HOUSLEY, R.; POLK, T. Planning for PKI. In: LONG, C. A. (Ed.). [S.l.]: Robert Ipsen, 2001.

IAIK. **Crypto Toolkits for the Java**. Institute for Applied Information Processing and Communication, 2013. Disponível em: <http://jce.iaik.tugraz.at/>.

IETF. **Public-Key Infrastructure (X.509) (pkix)**. 2013. Disponível em: <http://datatracker.ietf.org/wg/pkix/>.

International Telecommunication Union, I. T. U. Abstract Syntax Notation One. In: **International Telecommunication Union**. [s.n.], 2002. Disponível em: <http://www.itu.int/ITU-T/studygroups/com17/languages/>.

ITI. **Manual de Condutas Técnicas 7 - Procedimentos de Ensaios para Avaliação de Conformidade aos Requisitos Técnicos de Módulos de Segurança Criptográfica (MSC) no âmbito da ICP-Brasil**. [S.l.], set. 2007. v. 2, n. 2. Disponível em: <http://www.iti.gov.br/images/servicos/homologacao/MCT7_-_Vol.II.pdf>.

ITI. **Manual de Condutas Técnicas 7 - Requisitos, Materiais e Documentos Técnicos para Homologação de Módulos de Segurança Criptográfica (MSC) no Âmbito da ICP-Brasil**. [S.l.], set. 2007. v. 1, n. 1. Disponível em: <http://www.iti.gov.br/images/servicos/homologacao/MCT7_-_Vol.I.pdf>.

ITI. **Programa João de Barro**. Março 2008. Disponível em: <http://www.iti.gov.br/programas/programa-joao-de-barro>.

ITI. **Manual de Condutas Técnicas 11 - Procedimentos de Ensaios para Avaliação de Conformidade aos Requisitos Técnicos de Softwares de AC e AR no âmbito da ICP-Brasil**. [S.l.], 2010. v. 2, n. 2. Disponível em: <http://www.iti.gov.br/images/servicos/homologacao/MCT_11_2.pdf>.

ITI. **Manual de Condutas Técnicas 11 - Requisitos, Materiais e Documentos Técnicos para Homologação de Software de Autoridade Certificadora (AC) e Autoridade de Registro (AR) no Âmbito da ICP-Brasil**. [S.l.], 2010. v. 1, n. 1. Disponível em: <http://www.iti.gov.br/images/servicos/homologacao/MCT_11_1.pdf>.

JELLINGHAUS, A. et al. **libp11 - PKCS#11 wrapper library**. Git Hub, 2013. Disponível em: <https://github.com/OpenSC/libp11/wiki>.

KOHLER, J. G.; JUNIOR, R. B. **CONTRIBUIÇÕES AO SISTEMA DE GERENCIAMENTO DO CICLO DE VIDA DE CERTIFICADOS DIGITAIS DA INFRAESTRUTURA DE CHAVES PÚBLICAS PARA PESQUISA E ENSINO (SGCI)**. Tese (Trabalho de Conclusão de Curso) — Universidade Federal de Santa Catarina, 2007. Disponível em: <https://projetos.inf.ufsc.br/arquivos_projetos/projeto_691/SGCI2.pdf>.

KOHNFELDER, L. M. Towards a practical public-key cryptosystem. In: M.I.T., DEPT. OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE. **Archives**. Massachusetts Institute of Technology, 1978. Disponível em: <http://groups.csail.mit.edu/cis/theses/kohnfelder-bs.pdf>.

Kryptus. **AHX4 ASI-HSM**. Kryptus, 2013. Disponível em: <http://www.kryptus.com/>.

LabSEC. **LibCryptoSec**. Laboratório de Segurança em Computação, 2010. Disponível em: <https://projetos.labsec.ufsc.br/libcryptosec>.

LabSEC. **Sistema Gerenciador de Certificados Digitais da ICPEdu**. Março 2013. Disponível em: <https://projetos.labsec.ufsc.br/sgci>.

MARTINA, J. E.; SOUZA, T. C. S. de; CUSTODIO, R. F. Openhsm: An open key life cycle protocol for public key infrastructure's hardware security modules. In: **Public Key Infrastructure**. [S.l.]: Springer, 2007. p. 220–235.

MARíN, D. **gnoMint**. set. 2006. Disponível em: <http://gnomint.sourceforge.net/>.

MENTI, L. A. **Auditoria em Unidade Certificadora**. Tese (Trabalho de Conclusão de Curso) — Universidade Federal de Santa Catarina, jul. 2008. Disponível em: <https://projetos.inf.ufsc.br/arquivos_projetos/projeto_714/Auditoria%20em%20Unidade%20Certificadora%20-%20Leonardo%20Albuquerque%20Menti.pdf.1>.

Mozilla. **Network Security Services for Java (JSS)**. Mozilla, 2013. Disponível em: <https://developer.mozilla.org/en-US/docs/JSS>.

NYSTROM, M.; KALISKI, B. **PKCS #10: Certification Request Syntax Specification Version 1.7**. IETF, nov. 2000. RFC 2986 (Informational). (Request for Comments, 2986). Disponível em: <http://www.ietf.org/rfc/rfc2986.txt>.

OpenSSL. **OpenSSL Engine**. 2013. Disponível em: <http://www.openssl.org/docs/crypto/engine.html>.

Oracle. **Java API for RESTful Services (JAX-RS)**. Oracle, 2013. Disponível em: <http://jax-rs-spec.java.net/>.

Oracle. **Java Native Interface (JNI)**. Oracle, 2013. Disponível em: <http://docs.oracle.com/javase/6/docs/technotes/guides/jni/>.

PrimeKey. **EJBCA Enterprise PKI CA**. Agosto 2013. Disponível em: <http://www.ejbca.org/index.html>.

RIVEST, R.; SHAMIR, A.; ADLEMAN, L. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. In: **Communications of the ACM**. ACM, 1978. v. 21, n. 2, p. 120–126. Disponível em: <http://theory.lcs.mit.edu/~rivest/rsapaper.ps,>.

RNP. **Infraestrutura de Chaves Públicas para Ensino e Pesquisa (ICPEdu)**. 2013. Disponível em: <http://www.rnp.br/servicos/icpedu.html>.

RSA Laboratories. **PKCS #11: Cryptographic Token Interface Standard**. 2009. Disponível em: <http://www.rsa.com/rsalabs/node.asp?id=2133>.

SANTESSON, S. et al. **[RFC-6960] Internet X.509 Public Key Infrastructure - Online Certificate Status Protocol - OCSP**. IETF, jun. 2010. RFC 6960 (Informational). (Request for Comments, 6960). Disponível em: <http://www.ietf.org/rfc/rfc6960.txt>.

SCHAAD, J. **Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF)**. IETF, set. 2005. RFC 4211 (Proposed Standard). (Request for Comments, 4211). Disponível em: <http://www.ietf.org/rfc/rfc4211.txt>.

SCHAAD, J.; MYERS, M. **[RFC-5272] Certificate Management over CMS (CMC)**. IETF, jun. 2008. RFC 5272 (Informational). (Request for Comments, 5272). Disponível em: <http://tools.ietf.org/html/rfc5272>.

SCHAAD, J.; MYERS, M. **[RFC-5273] Certificate Management over CMS (CMC): Transport Protocols**. IETF, jun. 2008. RFC 5273 (Informational). (Request for Comments, 5273). Disponível em: <http://tools.ietf.org/html/rfc5273>.

SCHAAD, J.; MYERS, M. **[RFC-5274] Certificate Management Messages over CMS (CMC): Compliance Requirements**. IETF, jun. 2008. RFC 5273 (Informational). (Request for Comments, 5273). Disponível em: <http://tools.ietf.org/html/rfc5274>.

SCHNEIER, B.; KELSEY, J. Cryptographic support for secure logs on untrusted machines. In: **Proceedings of the 7th USENIX Security Symposium**. [S.l.: s.n.], 1998. p. 53–62.

SHAMIR, A. How to share a secret. In: **Communications of the ACM**. [s.n.], 1979. v. 22, n. 11, p. 612–613. ISSN 00010782. Disponível em: <http://portal.acm.org/citation.cfm?doid=359168.359176>.

SILVéRIO, A. L. **ANÁLISE E IMPLEMENTAÇÃO DE UM PROTOCOLO DE GERENCIAMENTO DE CERTIFICADOS**. Tese (Trabalho de Conclusão de Curso) — Universidade Federal de Santa Catarina, 2011. Disponível em: <https://projetos.inf.ufsc.br/arquivos_projetos/projeto_1163/TCC-Anderson-Final.pdf>.

SM-Zone. **TinyCA**. jul. 2006. Disponível em: <http://www.sm-zone.net/>.

STRAUB, T. Usability Challenges of PKI. **Doctor**, v. 5, n. November, p. 4–4, 2006. Disponível em:

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.110.1100\&rep=rep1\&type=pdf>.

SUN. **Java Cryptography Architecture**. 2002. Disponível em: <http://docs.oracle.com/javase/1.4.2/docs/guide/security/CryptoSpec.html>.

SUTIL, J. M. **GESTÃO SEGURA DE MÚLTIPLAS INSTÂNCIAS DE UMA MESMA CHAVE DE ASSINATURA EM AUTORIDADES CERTIFICADORAS**. Tese (Master Thesis) — Universidade Federal de Santa Catarina, 2011. Disponível em: <http://sgc.labsec.ufsc.br/~jeanms/dissertacao-jeandre.pdf>.

WERLANG, F. C.; MARTINS, L. G. **SISTEMA GERENCIADOR DE CERTIFICADO OFFLINE**. Tese (Trabalho de Conclusão de Curso) — Universidade Federal de Santa Catarina, 2010. Disponível em: <https://projetos.inf.ufsc.br/arquivos_projetos/projeto_985/TCC_SGC_Offline.pdf>.